Revisiting the Optimal PMU Placement Problem in Multi-Machine Power Networks

Mohamad H. Kazma, *Graduate Student Member, IEEE* and Ahmad F. Taha$^{\diamond}$, *Member, IEEE*

***Abstract*—To provide real-time visibility of physics-based states, phasor measurement units (PMUs) are deployed throughout power networks. PMU data enable real-time grid monitoring and control—and is essential in transitioning to smarter grids. Various considerations are taken into account when determining the geographic, optimal PMU placements (OPP). This paper focuses on the control-theoretic, observability aspect of OPP. A myriad of studies have investigated observability-based formulations to determine the OPP within a transmission network. However, they have mostly adopted a simplified representation of system dynamics, ignored basic algebraic equations that model power flows, disregarded including renewables such as solar and wind, and did not model their uncertainty. Consequently, this paper revisits the observability-based OPP problem by addressing the literature's limitations. A nonlinear differential algebraic representation (NDAE) of the power system is considered and implicitly discretized—using various different discretization approaches—while explicitly accounting for uncertainty. A moving horizon estimation approach is explored to reconstruct the joint differential and algebraic initial states of the system, as a gateway to the OPP problem which is then formulated as a computationally tractable integer program (IP). Comprehensive numerical simulations on standard power networks are conducted to validate various aspects of this approach and test its robustness to various dynamical conditions.**

***Index Terms*—Power system modeling, phasor measurements, nonlinear differential algebraic models, nonlinear observability, moving horizon estimation, optimal PMU placement**

I. Introduction, Motivation and Paper Contributions

A. Motivation

WITHIN recent years, there has been a trend of shifting from fuel-based energy generation to fuel-free, renewables. This transition has imposed a challenge on preserving stable grid operations [1]. To enable stable and secure power system operations, grid monitoring is performed by the supervisory and data acquisition (SCADA) system. [2]. However, SCADA measurements cannot fully capture the dynamics of the power system under transient conditions due to slow system update rates [3]–[5]. As such, more elaborate real-time system monitoring is required to perform state estimation and control. Dynamic state estimation (DSE) from real-time measurements allows for reconstructing system dynamics, thereby enabling stability predictions and control. In order to achieve an observable system for DSE, Phasor measurement units (PMUs) are installed within the grid in order to provide real-time high-resolution measurements [6]. Albeit a system can have PMUs allocated to each bus and achieve complete observability, this is not economical [7]. Thus, it is necessary to solve for optimal PMU placement (OPP) posed as a constraint optimization problem whilst achieving observability [4].

Previous studies have focused on the PMU placement problem considering steady state estimation (SSE). Nonetheless, such methods that are based on steady state operating conditions are unsuitable for power systems for-which exhibit considerable dynamic changes [8]. Moreover, SSE based sensor placement is mainly based on topological observability and therefore, neglects important parameters [5]. A comprehensive survey on the different approaches that deal with OPP under topological observability in power systems are presented in [9], [10].

Herein, we focus on formulating the OPP from an observability-theoretic perspective. Meaning that the placement problem is posed so that the system is observable and therefore, control and stability predictions under DSE is achievable [11]. Howbeit, current literature that has addressed the combinatorial observability-based PMU placement problem in nonlinear power networks establish that such sensor selection problem (i) is not well understood and is solved via a heuristics methods that becomes unfeasible for large networks [12] and (ii) is still considered an open problem for a nonlinear representation of the power system [13]—given the complexity of the nonlinearities that are evident in the observability analysis of the system.

B. Literature review

A primary step in developing an OPP program is quantifying the observability of the dynamic system. This quantification for nonlinear networks can be approached under several formulations. One approach is substituting the nonlinearities by first order linear approximations [14], [15]. This approach yields inaccurate observability-based analysis under uncertainties that change the operating conditions for which the linearized dynamics are valid. Other prevalent approaches include the use of Lie derivatives and differential embedding to construct the observability matrix [16], [17]. However, such formulations do not guarantee optimal sensor selection for network observability [13]. Furthermore, quantifying observability in nonlinear networks can be based on the empirical observability Gramian [18]–[20]. The positive semi-definite matrix structure of the empirical observability Gramian which relates energy notions of controllability and observability quantitative metrics—trace, determinant, rank, etc.—allows for sensor selection [12].

In [21], the optimal sensor placement for a nonlinear network is posed as a maximization of the empirical observability Gramian's determinant, however for systems that are marginally observable, this metric results in numerical problems. Moreover, the formulation results in a mixed integer nonlinear program which is computationally complex and nontrivial for large networks. Similarly for a nonlinear power system, the study [5]

$^{\diamond}$Corresponding author. This work is supported by National Science Foundation under Grants 2152450 and 2151571. The authors are with the Civil & Environmental Engineering and Electrical & Computer Engineering Departments, Vanderbilt University, 2201 West End Ave, Nashville, Tennessee 37235. Emails: mohamad.h.kazma@vanderbilt.edu, ahmad.taha@vanderbilt.edu.

poses the PMU selection problem based on empirical observability Gramian metrics and albeit good observability under the optimal PMU placement is obtained however, the OPP (i) is nonetheless performed under typical flow conditions and then the robustness of the optimal solution is examined, (ii) is limited to estimation of dynamic states and not the joint dynamic and algebraic state estimation, and (iii) is posed as a nonconvex mixed integer program that is computationally exhaustive. To tackle the computational complexity of the OPP, the authors in [12] approach solving the OPP problem by introducing the idea of leveraging observability Gramian metrics' submodularity properties. However, this approach yields sub-optimal placements given it was solved using a greedy approach.

Others studies [22]–[28] have approached the observability-based OPP problem based on the formulations similar to those introduced in the survey above [5], [12], [21], while also not considering all the aforementioned drawbacks. In power systems control, typically the differential equations are considered in the system representation of the model, whereas the algebraic equations are neglected due to the computational burden and overall stability implications [29]. A complete representation of a power system includes both differential and algebraic equations. The advantages of simulating the dynamics of the system under a complete nonlinear differential algebraic equations (NDAE) formulation are: (i) linking of network dynamics with power flow equations resulting in an accurate dynamics representation [30], (ii) modeling load and renewable uncertainties in DSE routines as a result of incorporating renewables in the system [31], and (iii) expanding the set of potential measurement buses to include non-generator buses.

C. Paper contributions and organization

Motivated by the aforementioned limitations, the objective of this work is to develop an OPP formulation for an NDAE representation of power system while (i) achieving full observability under uncertainty from loads and renewables, (ii) jointly estimating both dynamic and algebraic states of the transmission network and, and (iii) posing the OPP under a computationally efficient formulation. It is to the best of our knowledge that observability-based OPP in power systems that are represented as a NDAE has not yet been investigated.

Accordingly, we approach formulating the OPP problem on the basis of leveraging the modularity of the observability matrix. The main contributions of this work are as follows

- We introduce and validate a structure preserving transformation that retains the complete NDAE representation while achieving a nonlinear ordinary differential equations (ODE) formulation. By using this model—denoted as μ-NDAE—we show that the observability matrix can be defined for the NDAE power system considered. We also showcase this model for three different implicit discrete-time modeling methods: backward differential formula (BDF), backward Euler (BE) and trapezoidal implicit (TI).
- As a stepping stone for the OPP problem, we reconstruct the joint dynamic and algebraic initial states by adopting a moving horizon (MHE) framework. The state estimation is posed as a nonlinear least-squares problem which we solve numerically using the Gauss-Newton algorithm.
- We leverage the modularity property of the observability matrix to pose the OPP as a convex integer program (IP). Based on the modularity of the observability matrix, *a priori* observability information from each PMU placement is extracted prior to solving the OPP. Such approach extenuates the computational complexity of an optimization instance resulting in a computationally tractable approach for PMU placement in lager networks.
- The validity and effectiveness of this approach are studied on standard power networks. We show the validity of the μ-NDAE model under several discretizations and we prove that the optimal PMU placements for a specific number of PMUs are subsets of that for a larger number of PMUs, thus indicating modularity.

A preliminary and partial version of this paper appeared in [32] without proofs; it included a small case study on the viability of the proposed OPP approach. In this paper, we include several theoretical and numerical developments by (i) extending the proposed approach under 3 implicit discretization methods, (ii) providing detailed proofs and explicit Jacobian formulations for building the observability measures for each method, (iii) and extending the numerical studies to include OPP on larger power networks.

The remainder of this paper is organized as follows. In Section II, we introduce the NDAE power system and it's state-space formulation. In Section III, we present the different implicit discretizations of the NDAE system and the μ-NDAE system. In Section IV, initial state estimation based on a MHE approach is developed. In Section V, the OPP problem is formulated. The proposed OPP is studied for several standard power networks in Section VI. Finally, Section VII concludes the paper.

Paper's Notation: Let $\mathbb{N}$, $\mathbb{R}$, $\mathbb{R}^n$, and $\mathbb{R}^{p\times q}$ denote the set of natural numbers, real numbers, and real-valued row vectors with size of n, and p-by-q real matrices respectively. The symbol $\otimes$ denotes the Kronecker product. The cardinality of a set $\mathcal{N}$ is denoted by $|\mathcal{N}|$. The operators $\det(\boldsymbol{A})$ returns the determinant of matrix $\boldsymbol{A}$, $\mathrm{trace}(\boldsymbol{A})$ returns the trace of matrix of matrix $\boldsymbol{A}$ and $\mathrm{blkdiag}(\boldsymbol{A})$ constructs a block diagonal matrix.

II. Nonlinear power network DAE model

A power system $(\mathcal{N},\mathcal{E})$ can be represented graphically, where $\mathcal{E}\subseteq\mathcal{N}\times\mathcal{N}$ are the set of transmission lines, $\mathcal{N}=\mathcal{G}\cup\mathcal{L}$ is the set of all buses in the network, while $\mathcal{G}$ and $\mathcal{L}$ are the set of generator and load buses respectively.

In this work, a NDAE formulation of a power system is studied. We consider the standard two axis 4^{th} order transient model of a synchronous generator [33]. This model excludes exciter dynamics and turbine governor, meaning that each of the generators has 4 states and 2 control inputs. The dynamics of a synchronous generator $i\in\mathcal{G}$ can be written as (1)

$$\dot{\delta}_i = \omega_i - \omega_0 \tag{1a}$$

$$M_i\dot{\omega}_i = T_{\mathrm{M}i} - P_{\mathrm{G}i} - D_i(\omega_i-\omega_0) \tag{1b}$$

$$T'_{\mathrm{d0}i}\dot{E}'_i = -\frac{x_{\mathrm{d}i}}{x'_{\mathrm{d}i}}E'_i + \frac{x_{\mathrm{d}i}-x'_{\mathrm{d}i}}{x'_{\mathrm{d}i}}v_i\cos(\delta_i-\theta_i) + E_{\mathrm{fd}i} \tag{1c}$$

$$T_{\mathrm{CH}i}\dot{T}_{\mathrm{M}i} = T_{\mathrm{M}i} - \frac{1}{R_{\mathrm{D}i}}(\omega_i-\omega_0) + T_{\mathrm{r}i}, \tag{1d}$$

where the time varying components in (1) are: δ_i the rotor angle (rad), ω_i generator rotor speed $(\mathrm{rad/sec})$, $E_i^{'}$ generator transient voltage (pu), $T_{\mathrm{M}i}$ generator mechanical torque (pu). Generator inputs are: $E_{\mathrm{fd}i}$ generator internal field voltage (pu), $T_{\mathrm{r}i}$ governor reference signal (pu). Constants in (1) are: M_i is the rotor inertia constant $(\mathrm{pu} \times \mathrm{sec}^2)$, D_i is the damping coefficient $(\mathrm{pu} \times \mathrm{sec}^2)$, $x_{\mathrm{d}i}$ and $x_{\mathrm{q}i}$ are the direct-axis synchronous reactance (pu), $x_{\mathrm{d}i}^{'}$ is the direct-axis transient reactance (pu), $T_{\mathrm{d0}i}^{'}$ is the direct-axis open-circuit time constant (sec), $T_{\mathrm{CH}i}$ is the chest valve time constant (sec), $R_{\mathrm{D}i}$ is the speed governor regulation constant $(\mathrm{Hz/sec})$, and ω_0 is the synchronous speed $(120\pi \ \mathrm{rad/sec})$.

The algebraic constraints of the power system represent the relation between the internal states of a synchronous generator and it's generated power $P_{\mathrm{G}i}$ and $Q_{\mathrm{G}i}$ i.e, real and reactive power. The algebraic constraints of the nonlinear descriptor system can be written as (2) with $i \in \mathcal{G}$

$$P_{\mathrm{G}i} = \tfrac{1}{x_{\mathrm{d}i}^{'}} E_i^{'} v_i \sin(\delta_i - \theta_i) - \tfrac{x_{\mathrm{q}i} - x_{\mathrm{d}i}^{'}}{2x_{\mathrm{d}i}^{'} x_{\mathrm{q}i}} v_i^2 \sin(2(\delta_i - \theta_i)) \quad (2a)$$

$$Q_{\mathrm{G}i} = \tfrac{1}{x_{\mathrm{d}i}^{'}} E_i^{'} v_i \cos(\delta_i - \theta_i) - \tfrac{x_{\mathrm{q}i} - x_{\mathrm{d}i}^{'}}{2x_{\mathrm{d}i}^{'} x_{\mathrm{q}i}} v_i^2 - \tfrac{x_{\mathrm{q}i} - x_{\mathrm{d}i}^{'}}{2x_{\mathrm{d}i}^{'} x_{\mathrm{q}i}} v_i^2 \cos(2(\delta_i - \theta_i)). \quad (2b)$$

The power balance between the set of generator and load buses with $i \in \mathcal{G} \cup \mathcal{L}$ can be written as (3) such that, $N := |\mathcal{N}|$ is the number of buses within the transmission network while, $G := |\mathcal{G}|$ and $L := |\mathcal{L}|$ are the number of generator and load buses.

$$P_{\mathrm{G}i} + P_{\mathrm{L}i} = \sum_{j=1}^{N} v_i v_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) \quad (3a)$$

$$Q_{\mathrm{G}i} + Q_{\mathrm{L}i} = \sum_{j=1}^{N} v_i v_j (G_{ij} \cos\theta_{ij} - B_{ij} \sin\theta_{ij}), \quad (3b)$$

where $\theta_{ij} = \theta_i - \theta_j$ is the bus angle, v_i is the bus voltage (pu), (G_{ij}, B_{ij}) denote respectively the conductance and susceptance between bus i and j.

Having presented (1)–(3), which depict the physics based components of the electromechanical transients—representing both the generator dynamics and algebraic constraints—the state space formulation of the nonlinear descriptor power system can be written as (4)

$$\textit{generator dynamics}: \quad \dot{\boldsymbol{x}}_d = \boldsymbol{f}(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}) \quad (4a)$$

$$\textit{algebraic constraints}: \quad \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}_d, \boldsymbol{x}_a), \quad (4b)$$

where the dynamic states of the synchronous machine can be defined as $\boldsymbol{x}_d := \boldsymbol{x}_d(t) = [\boldsymbol{\delta}^\top \ \boldsymbol{\omega}^\top \ \boldsymbol{E}^{'\top} \ \boldsymbol{T}_{\mathrm{M}}^\top]^\top \in \mathbb{R}^{4G}$, the algebraic states can be defined as $\boldsymbol{x}_a := \boldsymbol{x}_a(t) = [\boldsymbol{P}_{\mathrm{G}}^\top \ \boldsymbol{Q}_{\mathrm{G}}^\top \ \boldsymbol{v}^\top \ \boldsymbol{\theta}^\top]^\top \in \mathbb{R}^{2G+2N}$ and the input of the system can be defined as $\boldsymbol{u} := \boldsymbol{u}(t) = [\boldsymbol{E}_{\mathrm{fd}}^\top \ \boldsymbol{T}_{\mathrm{r}}^\top]^\top \in \mathbb{R}^{2G}$. Matrix functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are nonlinear mapping functions such that, $\boldsymbol{f}(\cdot) : \mathbb{R}^{4G} \times \mathbb{R}^{2G} \times \mathbb{R}^{2G} \to \mathbb{R}^{4G}$ and $\boldsymbol{g}(\cdot) : \mathbb{R}^{4G} \times \mathbb{R}^{2G} \times \mathbb{R}^{2N} \to \mathbb{R}^{2G+2N}$. Based on the NDAE model of the power network presented above, the next section formulates the discrete-time model of the NDAE power system.

III. Implicit discrete-time modeling of power networks

In this section, we introduce the NDAE formulation referred to in this work as the μ-NDAE system. The impact of how the choice of discretization method is on the solution to the OPP is unclear and to that end we investigate the use of several implicit discrete-time modeling techniques and embed them with the OPP formulation.

DAE solvability have been thoroughly presented and investigated in literature. MATLAB is capable of solving DAEs with the DAE solvers—ode15i and ode15s [34]. However, given the discrete-time modeling approach that the observability-based OPP is herein based on, we refer to the use of numerical methods for simulating the NDAE power system. DAEs are considered unequivocally stiff and in particular, nonlinear power system models exhibit stiff dynamics [35]. Implicit discretization methods when used to simulate stiff dynamics offer a stable and computational efficient solution as compared with explicit discretization techniques. Implicit techniques previously used in the context of discrete-time modeling of power systems include: backward differential formulas (BDF) known as Gear's method [36], implicit Runge-Kutta (IRK) method [13], [37] and trapezoidal implicit (TI) method [38], [39]. The IRK method is the most numerically involved. BDF and TI methods have been shown to be an efficient methods for simulating power systems for transient stability analysis [40].

A. Discrete-time representation of NDAEs

With that in mind, we investigate the use of three implicit time-modeling methods (backward Euler (BE), TI, and BDF)—that account for the stiffness and complexity of the NDAEs—for solving the dynamics of system (4). Solving NDAEs using implicit numerical techniques requires finding a solution to a set of implicit nonlinear equations, which we implement using the Newton-Raphson (NR) method. In this section, we will showcase the discrete-time modeling approach for Gear's method.

Gear's k-step discretization method is generally stable for BDF discretization index k_g in the range of $2 \leq k_g \leq 5$. For $k_g = 1$, Gear's method represents BE. Thus, Gear's backward differential discretization is a generalization of BE's discretization. Accordingly, the discrete-time representation of (4) under Gear's method can be written as (5) for time step k with step size h, such that $\boldsymbol{x}_k := \boldsymbol{x}_{kh}$. We define vectors $\boldsymbol{z}_k := [\boldsymbol{x}_{d,k}, \ \boldsymbol{x}_{a,k}, \ \boldsymbol{u}_k]^\top$ and $\boldsymbol{x}_k := [\boldsymbol{x}_{d,k}, \ \boldsymbol{x}_{a,k}]^\top$ for time step k, and BDF discretization constant $\tilde{h} = \beta h$

$$\frac{\boldsymbol{x}_{d,k} - \sum_{s=1}^{k_g} \alpha_s \boldsymbol{x}_{d,k-s}}{\tilde{h}} = \boldsymbol{f}(\boldsymbol{z}_k) \quad (5a)$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}_k), \quad (5b)$$

where the term $\sum_{s=1}^{k_g} \alpha_s \boldsymbol{x}_{d,k-s}$ represents k_g previous time steps, and discretization constants β and α_s that depend on order of index k_g are calculated as

$$\beta = \Big(\sum_{s=1}^{k_g} \frac{1}{s} \Big)^{-1}, \quad \alpha_s = (-1)^{(s-1)} \beta \sum_{j=s}^{k_g} \frac{1}{j} \binom{j}{s}. \quad (6)$$

B. Structure-preserving μ-NDAE Representation

Before introducing the methodology under which we solve the NDAE system—also referred to as a descriptor system. We present a mathematical structural transformation to the NDAE,

that involves transforming the system in (4) from a NDAE into a nonlinear ODE representation. A descriptor system's index plays an important role in the complexity of the numerical simulation, whereby the higher an index, the more difficult it is to run the system [31]. The index of the NDAE system is related to it's algebraic equations and refers to the overall equivalency a NDAE has to an ODE [41]. The index-n of a NDAE system can be defined as 1.

Definition 1. *The descriptor system (1)-(3) is said to be of index-1 if, the DAEs can be converted into a system of ODEs by differentiating the system with respect to independent time variable (t) only once. That being said, the index-n of the descriptor system is the number of times needed to differentiate the DAEs to obtain system of ODEs.*

For the descriptor system (1)-(3), it can be shown that the system is of index-1 [30], [40]. The implicit function theorem can be used to transform system (4) from a DAE to an ODE structure [42]. Applying the aforementioned theorem and differentiating (4b) with respect to time we can obtain a NDAE model that is structurally equivalent to an ODE model and written as (7)

$$\dot{\boldsymbol{x}}_d = \boldsymbol{f}(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}) \tag{7a}$$

$$\dot{\boldsymbol{x}}_a = \tilde{\boldsymbol{g}}(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}) = -(\boldsymbol{G}_{\boldsymbol{x}_a})^{-1}\boldsymbol{G}_{\boldsymbol{x}_d}\boldsymbol{f}(\boldsymbol{x}_d, \boldsymbol{x}_a, \boldsymbol{u}), \tag{7b}$$

where matrix $\boldsymbol{G}_{\boldsymbol{x}_a} = \frac{\partial \boldsymbol{g}(\boldsymbol{x}_d, \boldsymbol{x}_a)}{\partial \boldsymbol{x}_a}$ and matrix $\boldsymbol{G}_{\boldsymbol{x}_d} = \frac{\partial \boldsymbol{g}(\boldsymbol{x}_d, \boldsymbol{x}_a)}{\partial \boldsymbol{x}_d}$.

With that in mind, we now discuss the rationale behind introducing an approximate transformation rather than the formulation presented in (7). We note here that the notion of transforming the NDAE system into an nonlinear ODE model is for reasons beyond numerical simulation and solvability. Howbeit a plethora of numerical methods have been developed to solve DAE systems particular DAEs of index-1 [43]. Herein, we are concerned with the aspect of observability for descriptor systems. In [44] the concept of algebraic observability is introduced, which formed a local observability definition for DAEs. The study related algebraic observability and local observability through a concept of regulating trajectory. This requires linearizing the NDAE system and writing it in an equivalent ODE system. Another study [45] tackled observer design within descriptor systems and formulated the concept of observability using Lie derivatives, however this was validated on a small scale system and was considered to be mathematically limited.

Granted that there is no conventional method in studying observability of NDAEs, we approach assessing the observability of a descriptor system (4) by representing the dynamics in an approximate ODE formulation that we refer to as μ-NDAE. Instead of using the implicit function theorem to represent the power system as (7)—that is computational expensive due to the existence of the partial derivative $\boldsymbol{G}_{\boldsymbol{x}_a}$ and its inverse—the left hand side is replaced in equation (4b) by $\mu\dot{\boldsymbol{x}}_a$ such that μ is a relatively small number which simulates the system's dynamics with a negligible error between the two representations while satisfying the power flow constraint equations. Therefore, the stability of the μ-NDAE is directly related to the value of μ chosen and does not depend on algebraic constraints $\boldsymbol{g}(\cdot)$, since $\boldsymbol{g}(\cdot)$ is of zero value. This means that the error that bounds the μ-NDAE is linearly proportional to the value of μ.

With the proposed approximation, the system is represented as an ODE, albeit without formulating unnecessary computations. The plausibility of such approximation is viable given the low index of the power system model—i.e, requiring one order of differentiation to become an ODE. The validity of such approximation is presented in Section VI for the implicit time-models under which the discretized dynamics are simulated.

Given such μ-NDAE approximation, the discrete-time representation of the power system in (5) can be rewritten in implicit form as (8) denoted by $\boldsymbol{\phi}(\boldsymbol{z}_k, \boldsymbol{x}_{k-s}) := \boldsymbol{\phi}(\boldsymbol{z}_k)$ such that,

$$\mathbf{0} = \boldsymbol{x}_{d,k} - \Sigma_{s=1}^{k_g}\alpha_s\boldsymbol{x}_{d,k-s} - \tilde{h}\boldsymbol{f}(\boldsymbol{z}_k) \tag{8a}$$

$$\mathbf{0} = \mu\boldsymbol{x}_{a,k} - \mu\Sigma_{s=1}^{k_g}\alpha_s\boldsymbol{x}_{a,k-s} - \tilde{h}\boldsymbol{g}(\boldsymbol{x}_k). \tag{8b}$$

C. NDAE numerical solvability: Newton-Raphson Method

The solvability of the discretized system in (8) involves finding a solution to a set of implicit nonlinear equations that is, finding $\boldsymbol{x}_d$ and $\boldsymbol{x}_a$ for each time step k. The NR method [40], [46] is implemented at each time-step to solve the set of equations under iteration index (i). The method is iterated until a relatively small error on the $\mathcal{L}_2$–norm of the iteration increment is achieved.

Based on such implicit nature of the μ-NDAE, we move forward with solving the system using NR method. First we represent (8) as (9) that is under iteration index (i) which depicts the convergence of NR's method. We denote (9) as $\boldsymbol{\phi}(\boldsymbol{z}_k^{(i)}, \boldsymbol{x}_{k-s}) := \boldsymbol{\phi}(\boldsymbol{z}_k^{(i)})$, where $\boldsymbol{z}_k^{(i)} := [\boldsymbol{x}_{d,k}^{(i)}, \boldsymbol{x}_{a,k}^{(i)}, \boldsymbol{u}_k^{(i)}]$ thereby retaining the same definition as $\boldsymbol{z}_k$ however now under the NR iteration index (i),

$$\mathbf{0} = \boldsymbol{x}_{d,k}^{(i)} - \Sigma_{s=1}^{k_g}\alpha_s\boldsymbol{x}_{d,k-s} - \tilde{h}\boldsymbol{f}(\boldsymbol{z}_k^{(i)}) \tag{9a}$$

$$\mathbf{0} = \mu\boldsymbol{x}_{a,k}^{(i)} - \mu\Sigma_{s=1}^{k_g}\alpha_s\boldsymbol{x}_{a,k-s} - \tilde{h}\boldsymbol{g}(\boldsymbol{x}_k^{(i)}). \tag{9b}$$

To ensure solution convergence for each time step, the Jacobian of the nonlinear dynamics in (8) is evaluated. Such that, at each time step k the increment $\Delta\boldsymbol{x}_k^{(i)}$—which is a function of the Jacobian—is evaluated and then is used to update state variable $\boldsymbol{x}_k^{(i+1)} = \boldsymbol{x}_k^{(i)} + \Delta\boldsymbol{x}_k^{(i)}$ for each iteration (i) until the convergence criterion is satisfied. Once NR method converges, time step k advances until the dynamics over time span (t) is simulated. The iteration increment $\Delta\boldsymbol{x}_k^{(i)}$ can be written as (10)

$$\Delta\boldsymbol{x}_k^{(i)} = \left[\boldsymbol{A}_g(\boldsymbol{z}_k^{(i)})\right]^{-1}\left[\boldsymbol{\phi}(\boldsymbol{z}_k^{(i)})\right], \tag{10}$$

where the Jacobian $\boldsymbol{A}_g(\boldsymbol{z}_k^{(i)}) = \left[\frac{\partial\boldsymbol{\phi}(\boldsymbol{z}_k^{(i)})}{\partial\boldsymbol{x}}\right]$ can be represented as (11)

$$\boldsymbol{A}_g(\boldsymbol{z}_k^{(i)}) = \begin{bmatrix} \boldsymbol{I}_{n_d} - \tilde{h}\boldsymbol{F}_{\boldsymbol{x}_d}(\boldsymbol{z}_k^{(i)}) & -\tilde{h}\boldsymbol{F}_{\boldsymbol{x}_a}(\boldsymbol{z}_k^{(i)}) \\ -\tilde{h}\boldsymbol{G}_{\boldsymbol{x}_d}(\boldsymbol{x}_k^{(i)}) & \mu\boldsymbol{I}_{n_a} - \tilde{h}\boldsymbol{G}_{\boldsymbol{x}_a}(\boldsymbol{x}_k^{(i)}) \end{bmatrix}. \tag{11}$$

We define $n_d := 4G$ as the number of differential states, $n_a := 2G + 2N$ as the number of algebraic states, $n := n_d + n_a$ as the number of differential and algebraic states. The matrix $\boldsymbol{F}_{\boldsymbol{x}_d} \in \mathbb{R}^{n_d \times n_d}$ represents the Jacobian of (9a) with respect to state variable $\boldsymbol{x}_d$, matrix $\boldsymbol{F}_{\boldsymbol{x}_a} \in \mathbb{R}^{n_d \times n_a}$ represents the Jacobian of (9a) with respect to algebraic variables $\boldsymbol{x}_a$, matrix $\boldsymbol{G}_{\boldsymbol{x}_d} \in \mathbb{R}^{n_a \times n_d}$ represents the Jacobian of (9b) with respect to state variables $\boldsymbol{x}_d$ and matrix $\boldsymbol{G}_{\boldsymbol{x}_a} \in \mathbb{R}^{n_a \times n_d}$ represents the Jacobian of (9b) with respect to algebraic variables $\boldsymbol{x}_a$. Matrix $\boldsymbol{I}_{n_d}$ is

an identity matrix of dimension similar to $\boldsymbol{F}_{\boldsymbol{x}_d}$ and $\boldsymbol{I}_{n_a}$ is an identity matrix of size similar to $\boldsymbol{G}_{\boldsymbol{x}_a}$.

The discrete-time models under BE and TI discretization methods are presented in A. The methodology for solving the NDAE representation of the power system that will be later used to validate the μ-NDAE approximation is presented in B.

IV. Initial State Estimation: A MHE Approach

In this section, we develop the framework for moving horizon state estimation (MHE) that is the basis of the OPP problem. Based on the discretized time-models developed in Section III, the discrete-time power system dynamics with measurements can be represented as

$$\boldsymbol{E}_\mu \boldsymbol{x}_k = \begin{cases} \boldsymbol{E}_\mu \boldsymbol{x}_{k-1} + \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\boldsymbol{z}_k) \\ \boldsymbol{g}(\boldsymbol{x}_k) \end{bmatrix} & \text{for BE,} \\ \boldsymbol{E}_\mu \sum_{s=1}^{k_g} \alpha_s \boldsymbol{x}_{k-s} + \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\boldsymbol{z}_k) \\ \boldsymbol{g}(\boldsymbol{x}_k) \end{bmatrix} & \text{for BDF,} \\ \boldsymbol{E}_\mu \boldsymbol{x}_{k-1} + \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\boldsymbol{z}_k) + \boldsymbol{f}(\boldsymbol{z}_{k-1}) \\ \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_{k-1}) \end{bmatrix} & \text{for TI,} \end{cases} \tag{12a}$$

$$\boldsymbol{y}_k = \tilde{\boldsymbol{C}}\boldsymbol{x}_k + \boldsymbol{\Gamma}\boldsymbol{v}_k, \tag{12b}$$

where $\boldsymbol{E}_\mu \in \mathbb{R}^{n\times n}$ is a diagonal matrix that has ones on its diagonal for $\boldsymbol{x}_d$ and μ for $\boldsymbol{x}_a$. Diagonal matrix $\boldsymbol{\Gamma}$ defines the placement of PMUs within the network such that $\boldsymbol{\Gamma} := \text{diag}(\gamma_z)$ and $\gamma_z = [0,1]^p$ whereby, $\gamma_z = 1$, if a PMU bus is selected and $\gamma_z = 0$, otherwise. Under such measurement model, we define $\mathcal{N}_p \subseteq \mathcal{N}$ as the set of buses at which PMUs can be installed, such that $|\mathcal{N}_p| := N_p$. We emphasize that since a full representation of the power system is being modeled—then, $N_p = N$—and thus includes both generator and non-generator buses. The matrix $\tilde{\boldsymbol{C}} := \boldsymbol{\Gamma}\boldsymbol{C} \in \mathbb{R}^{n_p \times n}$ represents the mapping of states variables under the selected PMU configuration. For the measurement model herein, $\tilde{\boldsymbol{C}}$ measures n_p states $[\boldsymbol{v}^\top \ \boldsymbol{\theta}^\top]^\top$ where $n_p := 2N_p$ represents the number of measured states. Variable $p \leq N_p$ denotes the number of selected PMUs within the transmission network and $\boldsymbol{v}_k \in \mathbb{R}^{n_p}$ is the measurement noise. Discretization constant $\tilde{h}$ for BE and TI discretization methods is defined in A.

Considering the discretized state-space measurement model with PMU placement presented in (12), we now introduce the MHE framework under which the observability-based OPP is postulated. The OPP program under the MHE framework utilized is based on the concept of observability for stiff nonlinear networks developed in [13]. The rationale behind referring to this approach is that it (i) adopts an simple open-loop MHE formulation, (ii) allows to study the influence of observation horizon window on state estimation accuracy, (iii) is intrinsically robust against measurement noise [47] and (iv) as compared with empirical observability Gramian and other approaches mentioned earlier in Section I, this method—as argued by [13]—is the most scalable approach for sensor selection within stiff nonlinear networks. Such approach has also been investigated on traffic networks applications, refer to [48].

To that end, we develop the observability based analysis through a MHE approach. To begin, we define an observation window equal to N_o discrete measurements. Then, we introduce a nonlinear vector function of the initial state $\boldsymbol{h}(\boldsymbol{\Gamma}, \boldsymbol{x}_0) := \boldsymbol{h}(\boldsymbol{x}_0) : \mathbb{R}^{n_p} \times \mathbb{R}^n \to \mathbb{R}^{n_p}$. The objective is to minimize the nonlinear least-square error on $\boldsymbol{h}(\cdot)$ which is posed as **P1**

$$(\mathbf{P1}) \quad \underset{\boldsymbol{x}_0}{\text{minimize}} \quad ||\boldsymbol{h}(\boldsymbol{x}_0)||_2^2 \tag{13a}$$

$$\text{subject to} \quad \underline{\boldsymbol{x}}_0 \leq \boldsymbol{x}_0 \leq \overline{\boldsymbol{x}}_0, \tag{13b}$$

where $\underline{\boldsymbol{x}}_0$ and $\overline{\boldsymbol{x}}_0$ are the lower and upper bounds on initial state variables. For power systems, the upper and lower bounds on algebraic variables are obtained from MATPOWER [49]. The vector function $\boldsymbol{h}(\cdot)$ represented in (14) is defined as $\boldsymbol{h}(\boldsymbol{x}_0) := \boldsymbol{y}(\boldsymbol{x}_0) - \boldsymbol{w}(\boldsymbol{\Gamma}, \boldsymbol{x}_0)$. Such that, the set of observations over N_o of the discretized μ-NDAE is represented by vector $\boldsymbol{y}(\boldsymbol{x}_0) \in \mathbb{R}^{n_p}$ and the nonlinear mapping vector function of the dynamics and algebraic states is represented by $\boldsymbol{w}(\boldsymbol{\Gamma}, \boldsymbol{x}_0) := \boldsymbol{w}(\boldsymbol{x}_0) : \mathbb{R}^{n_p} \times \mathbb{R}^n \to \mathbb{R}^{n_p}$, as such vector function $\boldsymbol{h}(\cdot)$ can be written as

$$\begin{bmatrix} \boldsymbol{h}(\boldsymbol{x}_0) \\ \boldsymbol{h}(\boldsymbol{x}_1) \\ \vdots \\ \boldsymbol{h}(\boldsymbol{x}_{\mathrm{N}_o-1}) \end{bmatrix} := \begin{bmatrix} \boldsymbol{y}_0 \\ \boldsymbol{y}_1 \\ \vdots \\ \boldsymbol{y}_{\mathrm{N}_o-1} \end{bmatrix} - \begin{bmatrix} \tilde{\boldsymbol{C}}\boldsymbol{x}_0 \\ \tilde{\boldsymbol{C}}\boldsymbol{x}_1 \\ \vdots \\ \tilde{\boldsymbol{C}}\boldsymbol{x}_{\mathrm{N}_o-1} \end{bmatrix}. \tag{14}$$

Remark 1. *The vector $\boldsymbol{g}(\cdot)$ in is in fact a function of initial state $\boldsymbol{x}_0 := \boldsymbol{x}(0)$, since the k-th state $\boldsymbol{x}_k := \boldsymbol{x}(k)$—as can be observed from (12b)—is coupled to initial state $\boldsymbol{x}_0$ through the postulated discrete state-space representation.*

Indeed for every initial condition $\boldsymbol{x}_0$, it holds true that $\boldsymbol{h}(\boldsymbol{x}_0) = 0$ such that, $\boldsymbol{y}(\boldsymbol{x}_0) = \boldsymbol{w}(\boldsymbol{x}_0)$. Therefore, we can define the observability of a system with respect to the selected PMU buses as Definition 2.

Definition 2. *Uniform observability of system (12) under the prescribed PMU placement holds true, if for all inputs $\boldsymbol{u}(k)$ and under a finite measurement horizon N_o, the mapping $\boldsymbol{g}(\cdot)$ defined in (14) is injective with respect to $\boldsymbol{x}_0$.*

That being said, we can say that for the system to be observable, initial state $\boldsymbol{x}_0$ under a selected sensor placement has to be uniquely determined for a set of measurements $\boldsymbol{y}(\boldsymbol{x}_0)$ over horizon N_o. A sufficient condition for $\boldsymbol{g}(\cdot)$ to be injective with respect to initial state $\boldsymbol{x}_0$, is that the Jacobian of $\boldsymbol{g}(\cdot)$ around $\boldsymbol{x}_0$ is full rank, that is, $\text{rank}(\boldsymbol{J}(\boldsymbol{\Gamma}, \boldsymbol{x}_0)) = n_d + n_a = n \ \forall \ \boldsymbol{x}_0$ [50].

A. Gauss-Newton for moving horizon estimation

Under such conditions, we can now solve the nonlinear least squares objective function (13) by exploiting the discrete nature of the system. We approach solving the least-squares optimization problem numerically using the Gauss-Newton (GN) algorithm. Such algorithm has been demonstrated on power systems for DSE [24], [51]. The reasons for referring to a numerical approach rather than utilizing already developed least-square solvers are two-fold. The first, is that GN algorithm is more computationally efficient and leads to solution converges faster and second, under the latter existing solvers approach while considering large networks—i.e, ACTIVSg200-bus case—MATLAB *lsqminorm* solver could not converge to an initial state estimate.

With that in mind, first to solve **P1** using GN we reformulate the objective and pose it as the minimization of the $\mathcal{L}_2$–norm of the residual function vector $\boldsymbol{r}(\boldsymbol{\Gamma}, \boldsymbol{q})$ that is concatenated from (i) the measurement equation (12b) and (ii) the discretized μ-NDAE model (12a). The redefined optimization problem is posed as **P2**

$$(\mathbf{P2}) \quad \underset{\boldsymbol{q}_0}{\text{minimize}} \quad ||\boldsymbol{r}(\boldsymbol{\Gamma}, \boldsymbol{q})||_2^2, \tag{15}$$

where the vector $\boldsymbol{q} \in \mathbb{R}^{\mathrm{N}_o \cdot n}$ is the concatenation of the dynamic and algebraic systems simulated over horizon N_o, whereby, can be written as $\boldsymbol{q} := [\boldsymbol{x}_{d,0}^\top,\ \boldsymbol{x}_{a,0}^\top,\ \ldots,\ \boldsymbol{x}_{d,\mathrm{N}_o-1}^\top,\ \boldsymbol{x}_{a,\mathrm{N}_o-1}^\top]^\top$. As such, the residual vector $\boldsymbol{r}(\boldsymbol{\Gamma}, \boldsymbol{q}) := \boldsymbol{r}(\boldsymbol{q}) \in \mathbb{R}^{\mathrm{N}_o \cdot n_p + \mathrm{N}_o \cdot n}$ is written as

$$\boldsymbol{r}(\boldsymbol{q}) := \begin{bmatrix} \boldsymbol{r_y} \\ \boldsymbol{r_x} \end{bmatrix}, \tag{16}$$

where vector $\boldsymbol{r_y} := \boldsymbol{h}(\boldsymbol{x}_0) = [\boldsymbol{r}_{\boldsymbol{y}_0}^\top \ \ldots \ \boldsymbol{r}_{\boldsymbol{y}_{\mathrm{N}_o-1}}^\top]^\top \in \mathbb{R}^{\mathrm{N}_o \cdot n_p}$ is the residual function of the measurement equation for N_o observations that is defined as (17), such that $\hat{\boldsymbol{x}}_k \in \mathbb{R}^{\mathrm{N}_o \cdot n}$ is the vector representing the estimated differential and algebraic states

$$\boldsymbol{r_y} := \boldsymbol{y}_k - \tilde{\boldsymbol{C}} \hat{\boldsymbol{x}}_k, \tag{17}$$

and vector $\boldsymbol{r_x} := \boldsymbol{\phi}(\boldsymbol{x}_0) = [\boldsymbol{r}_{\boldsymbol{x}_0}^\top \ \ldots \ \boldsymbol{r}_{\boldsymbol{x}_{\mathrm{N}_o-1}}^\top]^\top \in \mathbb{R}^{\mathrm{N}_o \cdot n}$ is the residual of the discretized μ-NDAE model, where $\boldsymbol{r}_{\boldsymbol{x}_k}$ for time step (k) is defined as

$$\boldsymbol{r}_{\boldsymbol{x}_k} := \begin{cases} \boldsymbol{E}_\mu(\hat{\boldsymbol{x}}_k - \hat{\boldsymbol{x}}_{k-1}) - \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\hat{\boldsymbol{x}}_k) \\ \boldsymbol{g}(\hat{\boldsymbol{x}}_k) \end{bmatrix} & \text{for BE,} \\ \boldsymbol{E}_\mu(\hat{\boldsymbol{x}}_k - \sum_{s=1}^{k_g} \alpha_s \hat{\boldsymbol{x}}_{k-s}) - \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\hat{\boldsymbol{x}}_k) \\ \boldsymbol{g}(\hat{\boldsymbol{x}}_k) \end{bmatrix} & \text{for BDF,} \\ \boldsymbol{E}_\mu(\hat{\boldsymbol{x}}_k - \hat{\boldsymbol{x}}_{k-1}) - \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\hat{\boldsymbol{x}}_k) + \boldsymbol{f}(\hat{\boldsymbol{x}}_{k-1}) \\ \boldsymbol{g}(\hat{\boldsymbol{x}}_k) + \boldsymbol{g}(\hat{\boldsymbol{x}}_k) \end{bmatrix} & \text{for TI.} \end{cases} \tag{18}$$

Having formed the residual function that is the objective of optimization problem **P2**, we move forward with solving the minimization problem using GN iterative method by updating state vector $\boldsymbol{q}$ such that (15) is minimized. The GN update for iteration (i) is given as (19) with a GN step size denoted by h_g.

$$\boldsymbol{q}^{(i+1)} = \boldsymbol{q}^{(i)} - h_g \big(\boldsymbol{J}_g(\boldsymbol{q}^{(i)})^\top \boldsymbol{J}_g(\boldsymbol{q}^{(i)})\big)^{-1} \boldsymbol{J}_g(\boldsymbol{q}^{(i)})^\top \boldsymbol{r}(\boldsymbol{q}^{(i)}). \tag{19}$$

The Jacobian matrix in (19) of the residual function $\boldsymbol{r}(\boldsymbol{q})$ is defined as (20)

$$\boldsymbol{J}_g(\boldsymbol{\Gamma}, \boldsymbol{q}^{(i)}) := \boldsymbol{J}_g(\boldsymbol{q}^{(i)}) = \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{N} \end{bmatrix}, \tag{20}$$

where Jacobian matrix of residual function $\boldsymbol{r_y}$ is denoted by $\boldsymbol{M}$ and defined as $\boldsymbol{M} := \text{blkdiag}(-\tilde{\boldsymbol{C}}) \in \mathbb{R}^{\mathrm{N}_o \cdot n_p \times \mathrm{N}_o \cdot n}$ while the Jacobian matrix of residual function $\boldsymbol{r_x}$ is denoted by $\boldsymbol{N}$ and defined as $\boldsymbol{N} := \text{blkdiag}(\boldsymbol{A}_g) \in \mathbb{R}^{\mathrm{N}_o \cdot n \times \mathrm{N}_o \cdot n}$. Here $\boldsymbol{A}_g \in \mathbb{R}^{n \times n}$ is the Jacobian of the discretized μ-NDAE (12a) which is evaluated for observation horizon N_o and is therefore dependent on the discretization method. Such that for BE and BDF discretization method the Jacobian matrix $\boldsymbol{A}_g$ is defined as (11) and for TI method $\boldsymbol{A}_g$ is defined in A. With the iteration update defined, Gauss-Newton iterative method is performed until the $\mathcal{L}_2$–norm of the residual (16) is minimized. Algorithm 1 outlines the proposed MHE for initial state estimation using GN method.

Algorithm 1: MHE via Gauss-Newton Iterations

Input: h_g, $\boldsymbol{x}_0$, $\boldsymbol{u}_0$, $\boldsymbol{\Gamma}$, tolerance
Output: $\hat{\boldsymbol{x}}_0$

```
Set i = 1 as GN iteration index
while L2–norm of the residual ≥ tolerance do
    Simulate the system dynamics with initial states x_0
    Build the residual function r(q) represented in (16)
    Calculate the Jacobian J_g(q^(i)) in (20)
    Perform the GN iteration update on q^(i+1) in (19)
    Update GN iteration index i = i + 1
    Update initial states x_0 → x̂_0
    Calculate L2–norm of the residual (16)
```

V. Observability-based PMU placement problem in power networks

In this section, we formulate the observability-based OPP that is based on the discretized system dynamics and MHE framework developed in Sections III and IV.

To quantify observability of the μ-NDAE representation of the power system, the concept of observability through the observability Gramian is used. Observability metrics that allow us to numerically quantify observability taking into account different aspects of the observability Gramian include: the condition number, rank, smallest eigenvalue, trace and determinant. Interested readers are referred to [5], [12] both of which presented a more elaborate discussion on the different metrics that quantify observability of the Gramian matrix. For the placement problem within the scope of this work, the trace of the observability Gramian is considered. The trace similar to the determinant quantifies the average observability in all directions of the state-space. The determinant is usually also considered since it is able to measure observability in the noise space. However, given the MHE approach that the placement formulation is built upon, redundancy towards noise is already considered prior to building the observability matrix.

Remark 2. *Additional consideration should be given if the observability Gramian has a large condition number, i.e, near zero eigenvalues. This implies that observability is ill-conditioned and that any perturbation to initial state $\boldsymbol{x}_0$ would change the observability rather significantly.*

As such, it is said that the observability Gramian is sensitive to uncertainties. Considering that, we pose OPP as a maximization of the trace of the observability matrix, while checking for near-zero eigenvalues. We implement the OPP on standard optimization interfaces such as YALMIP [52] along with Gurobi [53] solver. The OPP problem on the discretized state space measurement model (12) can be defined for a fixed number of sensors—denoted by p—as **P3**

$$(\mathbf{P3}) \quad \underset{\boldsymbol{\Gamma}}{\text{minimize}} \quad -\text{trace}\left(\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0)\right) \tag{21a}$$

$$\text{subject to} \quad \sum_{i=1}^{n} \gamma_i = p,\ \gamma_i \in \{0, 1\}^p, \tag{21b}$$

where $\boldsymbol{W}_o(\cdot) \in \mathbb{R}^{n \times n}$ is the observability Gramian of the nonlinear discretized μ-NDAE system. The observability Gramian

for a nonlinear descriptor system under a MHE formulation with PMU placement can be written as

$$\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0) = \boldsymbol{J}^T(\boldsymbol{\Gamma}, \boldsymbol{x}_0)\boldsymbol{J}(\boldsymbol{\Gamma}, \boldsymbol{x}_0), \tag{22}$$

where $\boldsymbol{J}(\cdot) \in \mathbb{R}^{\mathrm{N}_o \cdot n_p \times n}$ represents the Jacobian of function $\boldsymbol{h}(\cdot) = 0$ around $\boldsymbol{x}_0$ for the MHE observation horizon N_o and can be defined as

$$\boldsymbol{J}(\boldsymbol{\Gamma}, \boldsymbol{x}_0) := \left[\boldsymbol{I}_n \otimes \tilde{\boldsymbol{C}}\right] \begin{bmatrix} \boldsymbol{I}_n \\ \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \\ \vdots \\ \frac{\partial \boldsymbol{x}_{\mathrm{N}_o-1}}{\partial \boldsymbol{x}_0} \end{bmatrix}, \tag{23}$$

where

$$\frac{\partial \boldsymbol{x}_{j+1}}{\partial \boldsymbol{x}_j} = \frac{\partial \boldsymbol{x}_{j+1}}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j} \quad \text{for } j = \{0, 1, \ldots, \mathrm{N}_o - 1\}. \tag{24}$$

The observability formulation herein maps the sensor placement problem under a quantitative measure of DSE. That being said, the mapping of sensor location is represented by the matrix $\tilde{\boldsymbol{C}}$, where under full sensor placement—that is when all states are measured—the observability Gramian $\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0)$ is maximum, Hence, the objective function is minimum. Conversely, under zero sensing—that is $\tilde{\boldsymbol{C}}$ is equal to a zero matrix—a maximum error of zero is achieved.

As for calculating the Jacobian in (23), knowledge of $\boldsymbol{x}_k \ \forall \ j = \{1, \ldots, \mathrm{N}_o - 1\}$ is required. This can be obtained by simulating the discrete-time μ-NDAE dynamics over N_o. To calculate the partial derivatives terms in (23), that is $\frac{\partial \boldsymbol{x}_{j+1}}{\partial \boldsymbol{x}_j} = \frac{\partial \boldsymbol{x}_{j+1}}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}$ for $j = \{0, 1, \ldots, \mathrm{N}_o - 1\}$. We apply the chain rule to the j-th partial derivative as follows $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} \cdots \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0}$. However given the implicit nature of the discretized μ-NDAE system, the representation of $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_0}$ for the power system is not straightforward and depends on the discretization method followed [13]. For Gear's method we use the chain rule as presented in (25). The rationale behind the approach used for BDF method is described in D.

$$\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-k_g}} \cdots \frac{\partial \boldsymbol{x}_{k_g}}{\partial \boldsymbol{x}_0}. \tag{25}$$

Moreover, computing the Jacobian $\boldsymbol{J}(\cdot)$ for NDAEs is non-trivial. This is due to calculating the partial derivative of algebraic states whereby an explicit representation of the partial derivative $\frac{\partial \boldsymbol{x}_{a,j+1}}{\partial \boldsymbol{x}_{a,j}}$ for the algebraic states is non-trivial unless the system dynamics are reformulated into an ODE representation (7). For brevity we show this result in C. We note here that if we had referred to the use of the NDAE system instead of the approximate μ-NDAE representation—that retains an ODE structure—the process of expressing $\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}}$ in explicit form for the algebraic variable would have been non-trivial and hence the main rationale for such approximate transformation. With that in mind, we have opted to simulate the dynamics under the approximate μ-NDAE formulation presented in Section III. We present the Jacobian for the different discrete time-models (12a) that originate from the implicit state-space equations in C. In specific, we express the partial derivative $\frac{\partial \boldsymbol{x}_{j+1}}{\partial \boldsymbol{x}_j}$ in explicit form for each of the discrete time-models under study.

Having formulated the observability Gramian we now discuss reformulating the OPP problem **P3**. One approach for tackling the combinatorial class of sensor selection problems within networks, is posing such problem as a set function optimization problem where for a submodular[1] objective function, solving a set maximization problem is a common approach. Intrinsically, submodularity is considered to be a diminishing returns property [54]. Accordingly, the OPP in **P3** can be posed as a set function optimization program denoted by **P4**. The set of selected sensors is denoted by $\mathcal{Z} \subseteq \mathcal{N}_p$. The mapping of selected PMUs in set $\mathcal{Z}$ is encoded by the matrix $\tilde{\boldsymbol{C}}$.

$$(\mathbf{P4}) \quad \underset{\mathcal{Z}}{\text{minimize}} \quad -\operatorname{trace}\left(\boldsymbol{W}_o(\mathcal{Z}, \boldsymbol{x}_0)\right) \tag{26a}$$

$$\text{subject to} \quad |\mathcal{Z}| = p, \ \mathcal{Z} \subseteq \mathcal{N}_p. \tag{26b}$$

Submodular set maximization problem is still considered an NP-hard integer program. A common computationally tractable approach that achieves a sub-optimal solution for maximizing monotone increasing[2] submodular functions can be performed by a greedy heuristics approach. Solving the OPP under a greedy approach yields sub-optimal solutions that are at least $(1 - 1/e) = 63\%$ of the optimal solution [55].

Considering the above, we revisit the OPP posed in **P4** that is solved as a submodular set optimization program and instead pose it as an *a priori set optimization program*. The idea is based on the *a priori* observability knowledge from individual sensor measurements. The proposed framework involves computing prior singular contribution resulting from each PMU placement on the observability matrix. After saving such *a priori* information regarding observability contributions, the OPP that is then posed as a convex integer program (IP) is solved. The plausibility of such approach stems from the fact that the observability Gramian $\boldsymbol{W}_o(\cdot)$ is a modular[3] set function. In the context of linear systems, [12] showed that the observability matrix retains a modular set function structural property. With regards to the nonlinearities of model under study in this work, we prove that the observability matrix under PMU placement is modular with respect to to decision variable $\boldsymbol{\Gamma}$. The idea of considering the modularity of the observability Gramian is that a modular function forms positive linear combinations of the single elements in the modular set. This intuitively can be explained in the sense that modular function and linear functions are analogous whereby, each element within the set that forms the modular function has an independent contribution to the function value. With that in mind, the next proposition formulates the observability matrix $\boldsymbol{W}_o(\cdot)$ as a linear combination of it's individual elements.

Proposition 1. *The Observability matrix $\boldsymbol{W}_o(\cdot)$ can be written as a linear combination of individual observability matrices that are based on the individual contribution from each singular PMU placement as follows*

$$\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0) = \sum_{i=1}^{N_p} \boldsymbol{W}_{o,i}(\boldsymbol{\Gamma}_i, \boldsymbol{x}_0).$$

[1] A function $\mathcal{F} : 2^V \to \mathbb{R}$ is *submodular* if for every $A, B \subseteq V$, and $e \in V \backslash B$ it holds that $\Delta(e|A) \geq \Delta(e|B)$. Equivalently, a function $\mathcal{F} : 2^V \to \mathbb{R}$ is *submodular* if for every $A, B \subseteq V$ it holds that $\mathcal{F}(A \cap B) + \mathcal{F}(A \cup B) \leq \mathcal{F}(A) + \mathcal{F}(B)$

[2] A set function $\mathcal{F} : 2^V \to \mathbb{R}$ is monotone increasing if $\forall \ A, B \subseteq V$ the following holds true; $A \subseteq B \to \mathcal{F}(A) \leq \mathcal{F}(B)$

[3] A set function is modular if it is both submodular and supermodular, such that $\forall \ A, B \subseteq V$ the following holds true; $\mathcal{F}(A \cap B) + \mathcal{F}(A \cup B) = \mathcal{F}(A) + \mathcal{F}(B)$. Supermodularity of a set function $\mathcal{F}(\cdot)$ holds true if $-\mathcal{F}(\cdot)$ is submodular.

Accordingly, for each element in set $\mathcal{N}_p$ we evaluate, $\boldsymbol{W}_{o,i}(\boldsymbol{\Gamma}_i, \boldsymbol{x}_0)\ \forall\ i = \{1, \ \ldots, \ N_p\}$, prior to solving the OPP problem. The proof for Proposition 1—modularity of the observability matrix—is presented as follows.

Proof. First we consider $\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0)$ under full PMU placement that is $\tilde{\boldsymbol{C}} = \boldsymbol{C}$ then, the observability matrix can be written as

$$\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0) = \boldsymbol{J}^T(\boldsymbol{\Gamma}, \boldsymbol{x}_0)\boldsymbol{J}(\boldsymbol{\Gamma}, \boldsymbol{x}_0) \tag{27a}$$

$$= \begin{bmatrix} \boldsymbol{I}_n \\ \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \\ \vdots \\ \frac{\partial \boldsymbol{x}_{\mathrm{N}_o-1}}{\partial \boldsymbol{x}_0} \end{bmatrix}^\top \left[\boldsymbol{I}_n \otimes \boldsymbol{C}\right]^\top \left[\boldsymbol{I}_n \otimes \boldsymbol{C}\right] \begin{bmatrix} \boldsymbol{I}_n \\ \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \\ \vdots \\ \frac{\partial \boldsymbol{x}_{\mathrm{N}_o-1}}{\partial \boldsymbol{x}_0} \end{bmatrix}. \tag{27b}$$

We now reformulate (27) to show that it is analogous to a linear modular function. To do that, we refer to the distributive property of transpose over the Kronecker product, and thus we can write $\left[\boldsymbol{I}_n \otimes \boldsymbol{C}\right]^\top = \left[\boldsymbol{I}_n^\top \otimes \boldsymbol{C}^\top\right]$, then by using the mixed-product property of the Kronecker product we can write

$$\left[\boldsymbol{I}_n^\top \otimes \boldsymbol{C}^\top\right]\left[\boldsymbol{I}_n \otimes \boldsymbol{C}\right] = \underbrace{\left[\boldsymbol{I}_n^\top \boldsymbol{I}_n\right]}_{\boldsymbol{I}_n} \otimes \underbrace{\left[\boldsymbol{C}^\top \boldsymbol{C}\right]}_{\boldsymbol{C}_n}. \tag{28}$$

Replacing (28) in (27) and again with the use of the mixed-product property we can write

$$\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0) = \left[\begin{bmatrix} \boldsymbol{I}_n \\ \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \\ \vdots \\ \frac{\partial \boldsymbol{x}_{\mathrm{N}_o-1}}{\partial \boldsymbol{x}_0} \end{bmatrix}^\top \otimes \boldsymbol{C}_n\right]\left[\boldsymbol{I}_n \otimes \begin{bmatrix} \boldsymbol{I}_n \\ \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \\ \vdots \\ \frac{\partial \boldsymbol{x}_{\mathrm{N}_o-1}}{\partial \boldsymbol{x}_0} \end{bmatrix}\right], \tag{29}$$

we note here that matrix $\boldsymbol{C}_n \in \mathbb{R}^{n \times n}$ has 1's on the diagonal corresponding to the measured algebraic states and zeros elsewhere. As such $\boldsymbol{C}_n = \mathrm{blkdiag}(\boldsymbol{0}_{4G+2G}, \boldsymbol{I}_{2N})$. With that in mind, (29) can be rewritten as

$$\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0) = \begin{bmatrix} \boldsymbol{C}_n \\ \boldsymbol{C}_n \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \\ \vdots \\ \boldsymbol{C}_n \frac{\partial \boldsymbol{x}_{\mathrm{N}_o-1}}{\partial \boldsymbol{x}_0} \end{bmatrix}^\top \begin{bmatrix} \boldsymbol{I}_n \\ \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \\ \vdots \\ \frac{\partial \boldsymbol{x}_{\mathrm{N}_o-1}}{\partial \boldsymbol{x}_0} \end{bmatrix}, \tag{30}$$

where the only variable in the optimization matrix $\boldsymbol{W}(\cdot)$ is matrix $\tilde{\boldsymbol{C}} = \boldsymbol{C}_n$ which is a binary diagonal matrix mapping sensor locations within the power network. With that in mind, it is now evident that the observability matrix in (30) is linear with respect to to matrix $\boldsymbol{C}_n$.

As such, we define A as matrix $\boldsymbol{C}_n$ when bus no. 1 is measured and B as matrix $\boldsymbol{C}_n$ when bus no. 2 is measured. Then, $\boldsymbol{C}_n(A) = \mathrm{blkdiag}(\boldsymbol{0}_{4G+2G}, \boldsymbol{I}_2, \boldsymbol{0}_{2N-2})$ and $\boldsymbol{C}_n(B) = \mathrm{blkdiag}(\boldsymbol{0}_{4G+2G}, \boldsymbol{0}_2, \boldsymbol{I}_2, \boldsymbol{0}_{2N-4})$. Intuitively, the intersection of the two cases is zero and their union is their sum since $\boldsymbol{C}_n$ has only ones and zeros on it's diagonal. This concurs with the definition of modularity and thus the proof is complete. ■

Considering the above, the *a priori set optimization program* for optimal PMU placement denoted by **P5** can be posed as

$$(\mathbf{P5}) \quad \underset{\mathcal{Z}}{\text{minimize}} \quad -\mathrm{trace}\left(\boldsymbol{W}_o^1(\mathcal{Z}, \boldsymbol{x}_0)\right) \tag{31a}$$

$$\text{subject to} \quad |\mathcal{Z}| = p, \ \mathcal{Z} \subseteq \mathcal{N}_p, \tag{31b}$$

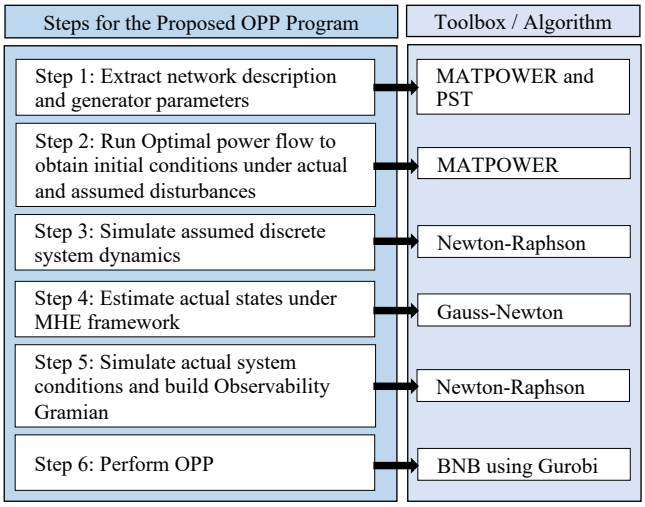


Figure 1. Implementation of optimal PMU placement framework for a NDAE representation of power systems.

where $\boldsymbol{W}_o^1(\mathcal{Z}, \boldsymbol{x}_0) = \sum_{i=1}^{N_p} \boldsymbol{W}_{o,i}(\mathcal{Z}_i, \boldsymbol{x}_0)$. Here $\mathcal{Z}_i$ corresponds to the selected i-th sensor that is encoded in matrix $\tilde{\boldsymbol{C}}$ that is, $\mathcal{Z}_i$ is a binary set that has a value of 1 at the i-th selected sensor location and zeros elsewhere, as such $|\mathcal{Z}_i| = 1$.

The concept of *a priori* optimization has been proposed before in optimization, in particular combinatorial optimization. Maros [56] introduced the concept of *a priori* optimization for optimizing randomly distributed networks in a computationally efficient manner. This concept encompasses attaining instance contributions knowledge prior to solving the combinatorial problem without having exponentially complex computations being performed during each optimization instance, i.e, at each optimization instance the complex computations are already evaluated. This allows one to perform combinatorial optimization with minimal computing power.

Having provided *a priori* information on an particular instance which in our case is possible given the modular nature of the observability matrix, **P5** which is categorized as a convex integer program (IP) is considered computational less exhaustive and therefore scalable to large power networks.

With such formulation, **P5** can be solved efficiently given that the observability metric (31a) is evaluating a linear combination of the individual pre-calculated contributions from each of selected PMUs. The implementation of this approach for OPP problem for a power system represented as a NDAE is summarized in Fig. 1. The validity and effectiveness of this approach is studied in the subsequent section of this paper.

VI. Case Studies: Validation and Results

In this section, we first validate the discrete-time μ-NDAE system developed in Section III and then we evaluate various aspects of the proposed OPP problem **P5**. The objective is to obtain an optimal PMU placement for a specified sensor fraction p that yields an observable system under load/renewables uncertainty. As such, we attempt to answer the following questions:

- **Q1**: What is an appropriate value for μ that offers a good compromise between numerical stability and accuracy in simulating system dynamics?

Table I
RSME VALUE OF THE SYSTEM STATES BETWEEN MATLAB ode15i AND THE DIFFERENT DISCRETIZATION METHODS UTILIZED TO SIMULATE THE NDAE POWER SYSTEM.

Network	Disturbance	RSME		
	α_L	BE	BDF	TRAP
case-9	2%	0.0022	1.2857×10^{-5}	0.0022
	3%	0.0049	2.0379×10^{-5}	0.0048
	4%	0.0126	9.5091×10^{-5}	0.0122
case-39	3%	0.2109	0.0134	0.1998
	5%	0.2171	0.0139	0.1908
	7%	0.2418	0.0172	0.2053
case-200	10%	0.0129	1.1396×10^{-5}	0.0131
	15%	0.0185	0.0010	0.0186
	20%	0.0227	0.0014	0.0228

- **Q2**: How does the choice of discretization method affect the power system simulation, and ultimately, the optimal PMU placements?
- **Q3**: Are the optimal PMU placements robust against load/renewables uncertainty and measurement noise?
- **Q4**: Does the framework under which we pose the OPP problem result in modular PMU placements and what is the significance of such modularity?

The simulations and optimization problem are performed in MATLAB R2021b running on a Macbook Pro having an Apple M1 Pro chip with a 10-core CPU and 16 GB of RAM. The PMU placement program is interfaced on MATLAB through YALMIP [52] and implemented using a standard brand and bound method (BNB) with Gurobi [53] as the solver.

We consider three different power networks for the assessment of the proposed approach:

- case-9: Western System Coordinating Council (WSCC) 9-Bus network (9-bus system with 3 synchronous generators).
- case-39: IEEE 39-Bus network "New-England Power System" (39-bus system with 10 synchronous generators).
- case-200: ACTIVSg200-Bus network "Illinois200 case" (200-bus system with 49 synchronous generators).

The test cases can be downloaded online from the Illinois center for a smarter electric grid cases repository [57]. The generator parameters are extracted from power systems toolbox (PST) [33] case file data3m9b.m and datane.m for case-9 and case-39 respectively. For case-200 the generator parameters are chosen based on the ranges provided in the PST toolbox. Regulation and chest time constants for the generators are chosen as $R_{\mathrm{D}i} = 0.2$ Hz/sec and $T_{\mathrm{CH}i} = 0.2$ sec, since they are not included in the PST case file. The steady state initial conditions for the power system are generated from the power flow solution obtained from MATPOWER [49]. The synchronous speed is set to $\omega_0 = 120\pi$ rad/sec and a power base of 100 MVA is considered for the power system.

To simulate the discretized descriptor system, we set the discretization step size $h = 0.1$ and simulations time $t = 30$ sec. Starting from the initial steady state conditions obtained from solving the power flow equations we introduce a load disturbance at $t > 0$ on initial load $(\mathrm{P}^0_{\mathrm{L}}, \mathrm{Q}^0_{\mathrm{L}})$. In this model renewables are modeled as a negative load, that is renewables are considered to inject power into the network as given in (3). The total power generation considered within the 3 cases is $(\mathrm{P}^0_{\mathrm{R}}, \mathrm{Q}^0_{\mathrm{R}}) = (0.2\mathrm{P}^0_{\mathrm{L}}, 0.2\mathrm{Q}^0_{\mathrm{L}})$. The perturbed magnitude under a load disturbance (α_L) is computed as $(\tilde{\mathrm{P}}^0_{\mathrm{L}}, \tilde{\mathrm{Q}}^0_{\mathrm{L}}) = (1 + \frac{\alpha_L}{100})(\mathrm{P}^0_{\mathrm{L}}, \mathrm{Q}^0_{\mathrm{L}})$. Moreover, the perturbed magnitude under a renewable disturbance (α_R) is computed as $(\tilde{\mathrm{P}}^0_{\mathrm{r}}, \tilde{\mathrm{Q}}^0_{\mathrm{r}}) = (1 + \frac{\alpha_R}{100})(\mathrm{P}^0_{\mathrm{R}}, \mathrm{Q}^0_{\mathrm{R}})$. Under the scope of this paper, we demonstrate simulating the system dynamics with load disturbance magnitude $\alpha_L)$ varying between $\{2\%, 20\%\}$ of the unperturbed initial loads and with renewable disturbance magnitude $\alpha_R = \alpha_L$ of the unperturbed initial renewable loads.

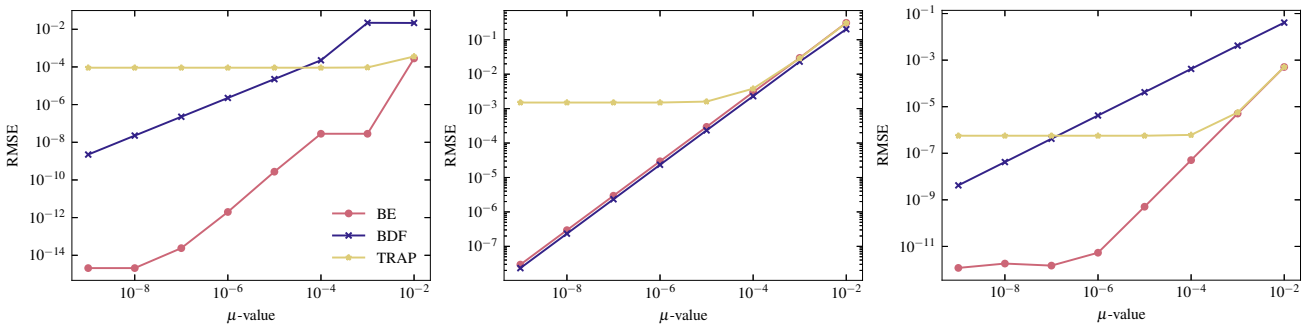
A. Simulating the discretized power system dynamics

To assess the accuracy of the discretization methods presented, we first simulate the baseline system dynamics using MATLAB DAE solver ode15i under the perturbations mentioned above. Then we simulate the dynamics using the discretization methods developed from Section III. Finally, we calculate the root mean square error (RSME) of the discretizations over time period t which is calculated as $\mathrm{RSME} := \sqrt{\frac{\sum_{k=1}^{t} \boldsymbol{e}_k^2}{t}}$ where $\boldsymbol{e}_k := |\tilde{\boldsymbol{x}}_k - \boldsymbol{x}_k|$ is the difference between the states of the two system representation with $\tilde{\boldsymbol{x}}_k$ corresponding to the discretized system and $\boldsymbol{x}_k$ to the system solved using ode15i. The setting chosen for ode15i are: (i) absolute tolerance as 1×10^{-05}, (ii) relative tolerance as 1×10^{-04} and (iii) maximum step size equal to 0.001. As for the Newton-Raphson algorithm we set: (i) absolute tolerance on $\mathcal{L}_2$–norm of iteration convergence as 10^{-2} and (ii) maximum iterations as 10. The results are summarized in Tab. I. It can be seen that for each of the three methods and under different load perturbations, the BDF discretization method outperforms BE and TI methods by having the lowest RSME values for the state estimates. The BDF discretization order chosen for the simulations is $k_g = 3$. Also, under increased load/renewable perturbations the systems results in larger a RSME value on the state trajectories, this is expected since the perturbations induce transient conditions within the system—that exhibit stiff nonlinearities and might render the system unstable.

B. Validating the discrete μ-NDAE model

We now move forward with validating the approximate μ-NDAE system. The validity of such approach is demonstrated for the three case systems by choosing different values for μ over a range of $\mu = \{10^{-2}, 10^{-9}\}$. We note that for any $\mu < 10^{-2}$ the μ-NDAE system does not converge to a solution, i.e, the power balance equations are not satisfied. That means for any value that is less than $\mu < 10^{-2}$, the simulation is unstable.

The RSME of the μ-NDAE model over time period t that is calculated by considering $\boldsymbol{x}_k$ corresponding to the NDAE system and $\tilde{\boldsymbol{x}}_k$ to the μ-NDAE system. The RMSE for the different cases and for each of the discretization methods over the range μ values is depicted in Fig. 2. It can be seen that the μ-NDAE system approximates the NDAE with a relatively small error and this error tends to decrease as μ approaches zero, which is intuitive since with μ reaching zero we go back



(a) (b) (c)

Figure 2. RMSE on dynamic and algebraic states between the NDAE and μ-NDAE discrete-time representations of the power systems: (a) case-9, (b) case-39, and (c) case-200.

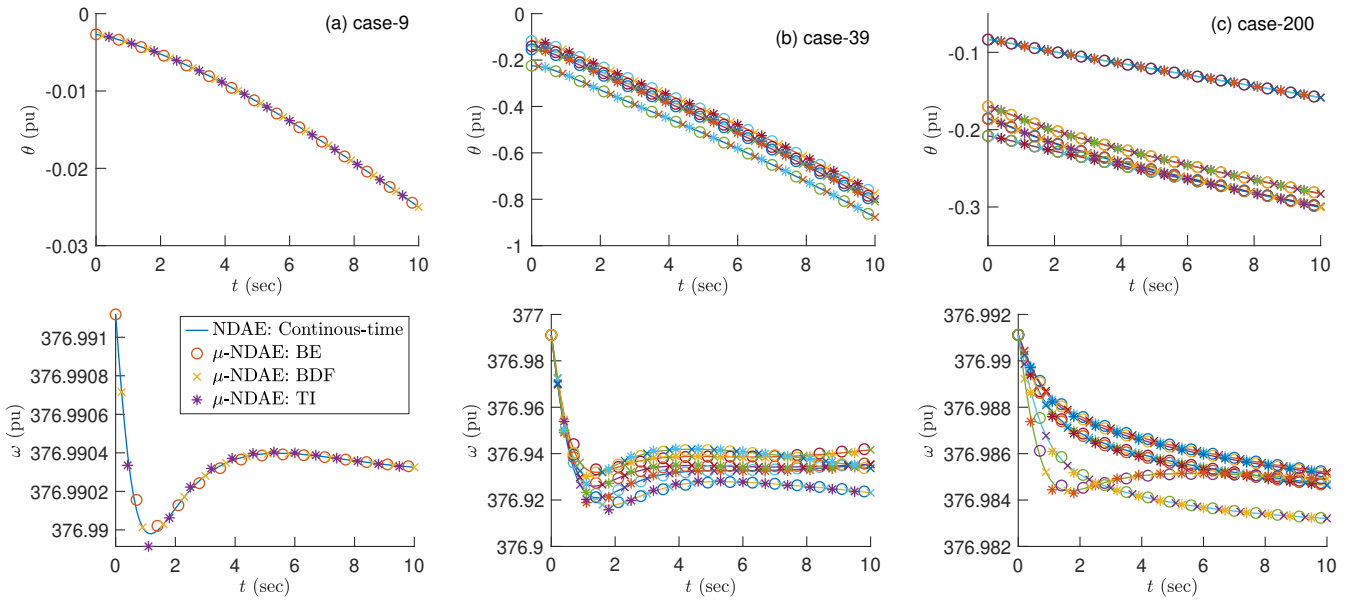


Figure 3. Transient differential (ω_i) and algebraic (θ_i) state trajectories under load and renewables disturbance: (a) case-9 ($\alpha_L = 2\%$), (b) case-39 ($\alpha_L = 5\%$) ,and (b) case-200 ($\alpha_L = 20\%$).

to having a NDAE. From Fig. 2 one can discern that the RSME becomes less than 10^{-3} when μ reaches 10^{-6} for the different discretization methods, in particular, that of TI where it has an asymptotic behavior when $\mu \geq 10^{-6}$. We note that, there is an approximately linear relation between the value of μ and the RMSE. This is evident in the BE and BDF cases, whereas it is asymptotically linear for TI case. This suggests that accuracy is directly related the value of μ. This means that the value of μ bounds the error on the resulting state trajectories under load/renewables disturbance, thus suggesting the following:

$$\sqrt{\sum_{k=1}^{t} \boldsymbol{e}_k^2} \overset{\approx}{\leq} \mu\sqrt{t} \tag{32}$$

Having provided experimental validation of the accuracy and stability of the proposed μ-NDAE model, for the remainder of this work we choose $\mu = 10^{-6}$ as it produces a sufficient approximation of the NDAE model. The differential and algebraic state trajectories of the studied power system cases under the different discretization methods are presented in Fig. 3 under $\mu = 10^{-6}$. The trajectories show accurate depiction of state-trajectories under the different discretization methods as compared with the baseline NDAE model.

C. Optimal PMU placement: under load/renewables uncertainty and measurement noise

We now solve the optimal PMU placement problem posed as **P5**, with an aim to seek an optimal configuration of PMU placement represented by set $\mathcal{Z}^*$ under a maximum number of PMUs denoted by p. The framework detailing the OPP program is presented in Fig. 1. To begin, we first initialize a power system under assumed initial conditions $\bar{\boldsymbol{x}}_0$ that has been perturbed under load and renewables disturbance $\alpha_R = \alpha_L = 4\%$, then by simulating the discretized measurement model in (12) and under $v = 2\%$ measurement noise over observation horizon N_o, we perform initial state estimation assuming full PMU placement—that is $|\mathcal{Z}| = n_p$. The GN method developed in Section IV for the MHE is implemented to solve for initial state estimate $\hat{\boldsymbol{x}}_0$ under optimization problem **P2**. As for the GN algorithm constants, we set time step constant $h_g = 0.1$ and tolerance on residual as 10^{-4}. Then, based on the initial state estimate the optimization problem **P5** is solved to obtain optimal set $\mathcal{Z}^*$ and compute the estimation error resulting from the optimal PMU placement. The estimation error that is based on the estimate of the GN algorithm is computed as $\varepsilon := \frac{\|\hat{\boldsymbol{x}}_0 - \boldsymbol{x}_0\|_2}{\|\boldsymbol{x}_0\|_2}$, where $\boldsymbol{x}_o$ is the actual state that we want to estimate and $\hat{\boldsymbol{x}}_o$

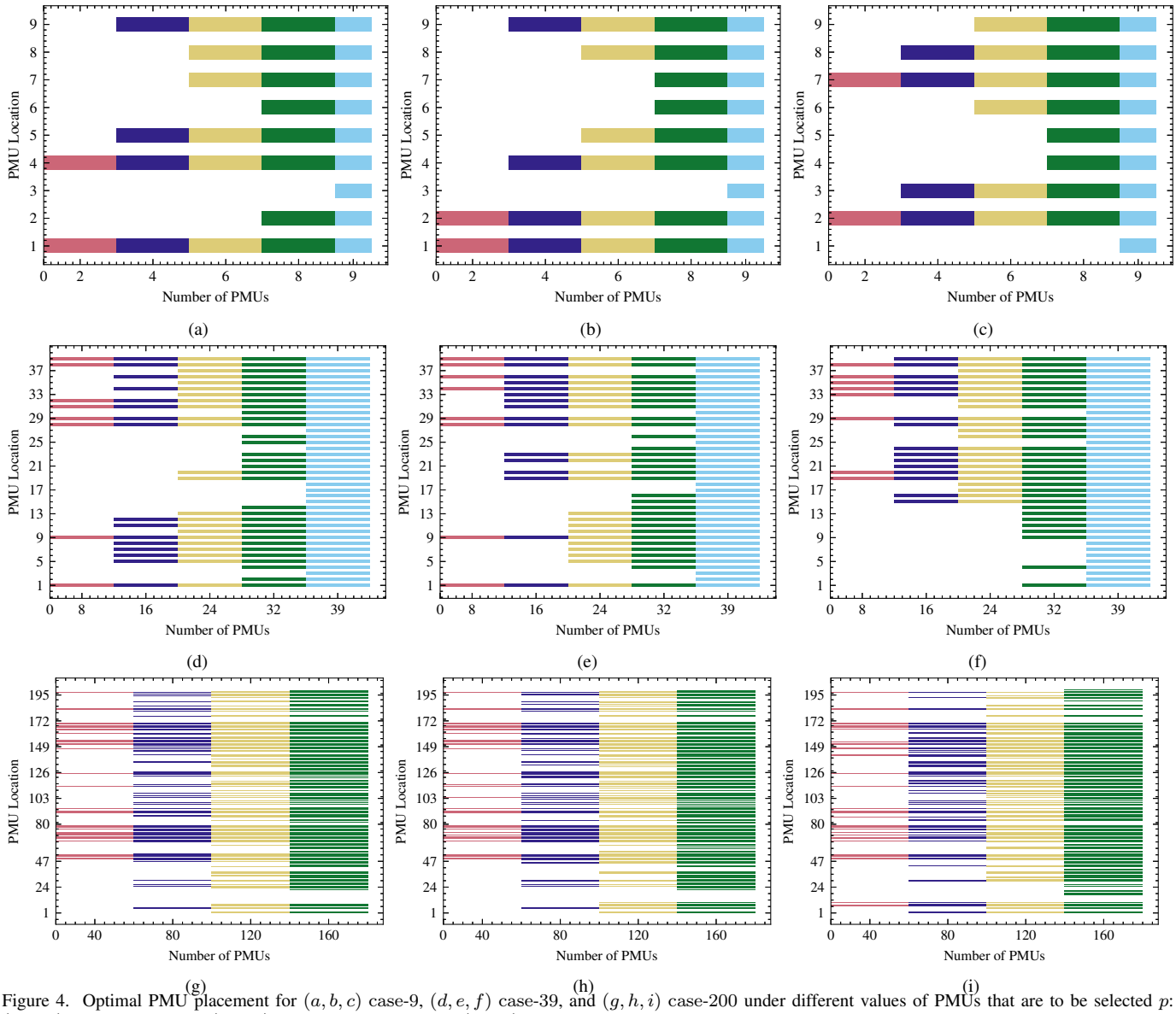


Figure 4. Optimal PMU placement for (a, b, c) case-9, (d, e, f) case-39, and (g, h, i) case-200 under different values of PMUs that are to be selected p: (a, d, g) BE discretization, (b, e, h) BDF discretization, and (c, f, i) TI discretization. The figures show a horizontal bar when a PMU is selected for each of the sensor fractions simulated.

is its estimate computed by solving the nonlinear least squares problem for the fixed sensor location. It is noteworthy to mention that **P5** is classified as a convex integer program (IP) since the presumed initial state estimate $\hat{\boldsymbol{x}}_0$ is fixed and binary vector $\boldsymbol{\Gamma}$ is the optimization variable.

We solve the OPP problem for each of test cases and under the different discretization methods while being constrained by the number of PMUs (p) that is to be employed within the network. The maximum number of PMUs to be installed for each of the test cases is taken as $p = \{0.2, 0.4, 0.6, 0.8\} \times n_p$. That is for example for case-9 we have $n_p = 9$, then for $p = 0.2 \times n_p$ we want to employ 2 PMUs within the network. The optimal PMU placements over the generator and load buses node locations for the each of the test cases and under different discretization methods are given in Fig. 4.

Three key aspects can be pointed out from the observability-based PMU placement program solved. The first is through the coupling of dynamics and algebraic states, load buses are selected and thus are included in the optimal set $\mathcal{Z}^*$. This is important since typically only generator buses are potential locations where PMUs can be installed under the observability-based approach for ODE power systems. Thus, validating the use of an NDAE representation of a power system instead of and ODE one for the observability-based OPP problem. The second is that the different discretization methods that the OPP is built upon yield different placements. This can be clearly identified from Figs. 4a, 4b, and 4c for case-9. The reasons for such behavior is that each of the discretization methods change the structure of the observability matrix (22). This is a result of the partial derivatives that have been derived and are presented for each of the discretization methods in C and D. One can notice that the TI method differs from both BDF and BE by having an additional evaluation of the partial derivative of the system nonlinearities for an additional previous time-step. Whereas BE and BDF differ by having k_g order of previous time step dependency. We point out that the placements for BE and

BDF are less different than that as compared to TI method. This similarity between BE and BDF OPP can be observed for case-200. The results suggest that indeed, the discretization method does affect the optimal PMU placements. We note here, that nonlinear models of observability depend on the operating point and the simulation model. That is, the nonlinear observability Gramian depends on the underlying structure of nonlinearities that are depicted in a dissimilar manner amongst the different discretizations and hence the different placements. In order to understand and assess the resulting optimal placements for each of the methods we compare the resulting estimation error on both differential and algebraic states in Section VI-D.

The third aspect is that it is evident that modularity is retained with the increase of PMUs selected. As depicted in Fig.4 there exists continuity of the horizontal histograms—that show the selected PMUs—as the number of PMUs selected increases. This means that as we increase the number of PMUs p required to be employed within the network, the same optimal sets $\mathcal{Z}^*$ for the previously specified p becomes as subset of the new OPP. This concurs with the modularity concept that the *a priori* optimization problem **P5** is based upon. The significance of having a modular PMU or sensor placement framework is two-folds. First, (i) with increased penetration of fuel-free energy sources—wind plants and solar farms—achieving an observable system simply requires the same grid phasor measurements or an additional PMU that augments the preexisting PMUs. This enables expanding grid operations while retaining system observability and control. Second, (ii) the scheduling of PMUs or sensors can be easily performed by activating an incremental set of sensors $\Delta\mathcal{Z}^*$. This offers a fast selection approach when selecting sensors or PMUs for DSE under physical constraints, such as cost or availability of sensors—the applicability of such points is to be explored in future work under this observability-based approach. On such note, we now asses the robustness of the optimal PMU placements against load/renewable perturbations and measurement noise.

1) Effect of measurement noise: To investigate the impact of measurement noise v on the optimal PMU placements, we vary v within the range of $v = \{0\%, 5\%\}$ and under each value we perform OPP for each of the sensor fractions p. The resulting placements show robustness towards measurement noise meaning that for each case, the same optimal placements were obtained. This can be explained by how the MHE estimation framework accounts for noisy measurements, such that the observability matrix is based on estimated measurements under noise from the MHE algorithm.

2) Effect of load/renewables uncertainty: As for investigating the impact of load/renewables uncertainty on the optimal placements, we vary α_L for case-9 and case-39 within the range of $\alpha_L = \{0\%, 5\%\}$ and for case-200 within the range of $\alpha_L = \{0\%, 20\%\}$. The results also show consistent placements with varying uncertainty on loads and renewables. This is also explained by the framework that the OPP is based upon whereby the major assumption is that initial states and load/renewables disturbances are not known and thus we start by assuming such conditions. Then, using the MHE Gauss-Newton algorithm the actual states under the actual loads/renewables are estimated and on those estimated states the observability matrix is constructed. This means that inherent within the construction of the observability matrix, such uncertainty is already accounted for and therefore, offers robustness towards load/renewables perturbations.

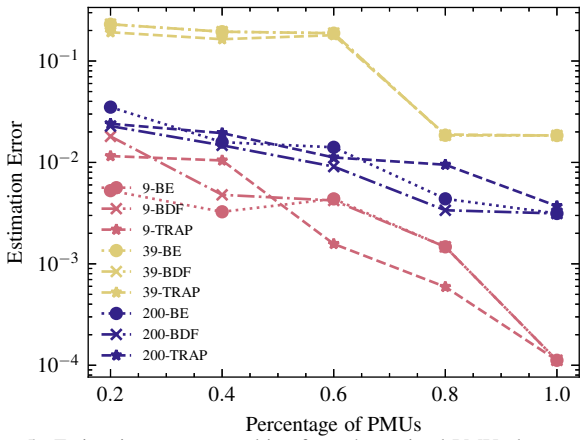


Figure 5. Estimation error ε resulting from the optimal PMU placements for each case and discretization methods.

D. Initial state estimation under optimal PMU placement

We perform initial state estimation based on the optimal PMU configuration chosen for each of the cases. The estimation error for each case and under each of the discretization methods is presented in Fig. 5. Intuitively, as we increase the number of PMUs placed, the estimation error decreases. This relates to the concept of observability whereby, the more PMUs are employed or more nodes being sensed, reconstruction of the initial dynamics and algebraic states becomes more accurate. This can be seen for each of the network cases in Fig. 5 as the ratio of PMUs—sensed nodes— is increased. For the case of PMUs placed that is between $\{0.2, 0.4\}$, BDF discretization estimation outperforms those of BE and TI. However, TI method becomes results in better estimated under increased PMU fractions. Herein, since we want to limit the number of PMUs, we consider the fractions that are small, i.e., employ lower PMUs whilst achieving adequate DSE. Thus based on the concept of observability and initial state reconstruction, we refer to BDF discretization method to solve for optimal PMU placements.

VII. Paper Summary and Future Work

This paper revisits the optimal PMU placement problem for power systems. The power system is based on a NDAE representation which allows coupling of the differential and algebraic states within the network. The NDAE system is discretized using BE, BDF, and TI discretization method and is transformed into a μ-NDAE which retains the mathematical structure of an ODE. We adopt a MHE approach to perform the OPP problem by exploiting the modularity of the observability matrix. As such, we pose the OPP as an *a priori set optimization program* which extenuates the computational burden from performing complex computation at each optimization instance of the combinatorial placement problem. Given the comprehensive computational investigation and validation, we answer the posed research questions in Section VI-C:

- **A1**: There is a value for μ the ensures numerical stability and solvability of the NDAE power system. Such value bounds the error on differential and algebraic state trajectories.
- **A2**: The choice of discretization method is important when formulating the nonlinear observability-based approach. This depends on the system nonlinearities and their structure. For the power system herein, BDF method is suggested for the OPP problem.
- **A3**: Indeed, robustness against measurement noise and load/renewables uncertainty is achieved under the OPP framework. This is inherent with the MHE framework that the observability-based OPP is built upon.
- **A4**: Modularity of the optimal PMU placements is observed when increasing the number of PMUs to be employed within the network. As mentioned in Section VI-C, this offer computationally efficient solutions for ever-growing power grids and sensor scheduling applications.

APPENDIX A
IMPLICIT DISCRETE-TIME MODELING

In this section, we present the backward Euler and trapezoidal implicit discrete-time models. We define the discretization constants $\tilde{h}$ for each of the method as follows

$$\tilde{h} := \begin{cases} h & \text{for BE} \\ \beta h & \text{for BDF} \\ 0.5h & \text{for TI.} \end{cases}$$

1) Backward Euler: Being the case that Gear's method is a generalization of backward Euler's discretization method, we can take the index $k_g = 1$. Then, the μ-NDAE system can be discretized under BE as

$$\boldsymbol{x}_{d,k} - \boldsymbol{x}_{d,k-1} = \tilde{h}\big(\boldsymbol{f}(\boldsymbol{z}_k)\big) \tag{33a}$$

$$\mu\boldsymbol{x}_{a,k} - \boldsymbol{x}_{a,k-1} = \tilde{h}\big(\boldsymbol{g}(\boldsymbol{x}_k)\big), \tag{33b}$$

and the system dynamics under BE method of the μ-NDAE system in a state-space representation can be written as

$$\boldsymbol{E}_\mu \boldsymbol{x}_k = \boldsymbol{E}_\mu \boldsymbol{x}_{k-1} + \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\boldsymbol{z}_k) \\ \boldsymbol{g}(\boldsymbol{x}_k) \end{bmatrix}, \tag{34}$$

such that the implicit representation of the BE discrete system (33) can be rewritten as

$$\boldsymbol{0} = \boldsymbol{x}_{d,k} - \boldsymbol{x}_{d,k-1} - \tilde{h}\boldsymbol{f}(\boldsymbol{z}_k) \tag{35a}$$

$$\boldsymbol{0} = \mu\boldsymbol{x}_{a,k} - \mu\boldsymbol{x}_{a,k-1} - \tilde{h}\boldsymbol{g}(\boldsymbol{x}_k). \tag{35b}$$

2) Trapezoidal Implicit: For the TRAP discretization method, the μ-NDAE system can be written as

$$\boldsymbol{x}_{d,k} - \boldsymbol{x}_{d,k-1} = \tilde{h}\big(\boldsymbol{f}(\boldsymbol{z}_k) + \boldsymbol{f}(\boldsymbol{z}_{k-1})\big) \tag{36a}$$

$$\boldsymbol{x}_{a,k} - \boldsymbol{x}_{a,k-1} = \tilde{h}\big(\boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_{k-1})\big), \tag{36b}$$

and the system dynamics under TI method of the μ-NDAE system in a state-space representation can be written as

$$\boldsymbol{E}_\mu \boldsymbol{x}_k = \boldsymbol{E}_\mu \boldsymbol{x}_{k-1} + \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \boldsymbol{f}(\boldsymbol{z}_k) + \boldsymbol{f}(\boldsymbol{z}_{k-1}) \\ \boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_{k-1}) \end{bmatrix}, \tag{37}$$

such that the implicit representation of the TI discrete system (33) can be rewritten as

$$\boldsymbol{0} = \boldsymbol{x}_{d,k} - \boldsymbol{x}_{d,k-1} - \tilde{h}\big(\boldsymbol{f}(\boldsymbol{z}_k) + \boldsymbol{f}(\boldsymbol{z}_{k-1})\big) \tag{38a}$$

$$\boldsymbol{0} = \mu\boldsymbol{x}_{a,k} - \mu\boldsymbol{x}_{a,k-1} - \tilde{h}\big(\boldsymbol{g}(\boldsymbol{x}_k) + \boldsymbol{g}(\boldsymbol{x}_{k-1})\big). \tag{38b}$$

Following such discretization schemes, we move forward with solving the system using Newton-Raphson analogously to that of Gear's method with the exception of having different Jacobian matrices under the iteration increment update.

The Jacobian $\boldsymbol{A}_g(\cdot)$ of the implicit BE-discretized nonlinear system has the same representation as of that of Gear's method. On the other hand, the Jacobian $\boldsymbol{A}_g(\cdot)$ of the implicit TI-discretized nonlinear system can be written as (39)

$$\boldsymbol{A}_g(\boldsymbol{z}_k^{(i)}) = \begin{bmatrix} \boldsymbol{I}_{n_d} - \tilde{h}\tilde{\boldsymbol{F}}_{\boldsymbol{x}_d}(\boldsymbol{z}^{(i)}) & -\tilde{h}\tilde{\boldsymbol{F}}_{\boldsymbol{x}_a}(\boldsymbol{z}^{(i)}) \\ -\tilde{h}\tilde{\boldsymbol{G}}_{\boldsymbol{x}_d}(\boldsymbol{x}^{(i)}) & \mu\boldsymbol{I}_{n_a} - \tilde{h}\tilde{\boldsymbol{G}}_{\boldsymbol{x}_a}(\boldsymbol{x}^{(i)}) \end{bmatrix}, \tag{39}$$

where $\tilde{\boldsymbol{F}}_{\boldsymbol{x}_d}(\boldsymbol{z}^{(i)}) := \boldsymbol{F}_{\boldsymbol{x}_d}(\boldsymbol{z}_k^{(i)}) + \boldsymbol{F}_{\boldsymbol{x}_d}(\boldsymbol{z}_{k-1}^{(i)})$, $\tilde{\boldsymbol{F}}_{\boldsymbol{x}_a}(\boldsymbol{z}^{(i)}) := \boldsymbol{F}_{\boldsymbol{x}_a}(\boldsymbol{z}_k^{(i)}) + \boldsymbol{F}_{\boldsymbol{x}_a}(\boldsymbol{z}_{k-1}^{(i)})$, $\tilde{\boldsymbol{G}}_{\boldsymbol{x}_d}(\boldsymbol{x}^{(i)}) := \boldsymbol{G}_{\boldsymbol{x}_d}(\boldsymbol{x}_k^{(i)}) + \boldsymbol{G}_{\boldsymbol{x}_d}(\boldsymbol{x}_{k-1}^{(i)})$, and $\tilde{\boldsymbol{G}}_{\boldsymbol{x}_a}(\boldsymbol{x}^{(i)}) := \boldsymbol{G}_{\boldsymbol{x}_a}(\boldsymbol{x}_k^{(i)}) + \boldsymbol{G}_{\boldsymbol{x}_a}(\boldsymbol{x}_{k-1}^{(i)})$.

APPENDIX B
SIMULATING THE IMPLICIT DISCRETE-TIME MODELS

In this section, we formulate the numerical methodology for solving the discrete-time NDAE system that will be used as a basis to validate the μ-NDAE system under which the OPP problem is posed.

Applying Gear's method to the continuous time NDAE system (4), the dynamics of the system concatenated as a state-space form can be rewritten as (40)

$$\boldsymbol{E}\boldsymbol{x}_k = \boldsymbol{E}\sum_{s=1}^{k_g} \alpha_s \boldsymbol{x}_{d,k-s} + \begin{bmatrix} \tilde{h} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{f}(\boldsymbol{z}_k) \\ \boldsymbol{g}(\boldsymbol{x}_k) \end{bmatrix}, \tag{40}$$

where $\boldsymbol{E} \in \mathbb{R}^{n\times n}$ is a singular matrix of $\text{rank} = n_d$ and is defined as

$$\boldsymbol{E}(i,i) := \begin{cases} 1 & \text{for} \quad 1 \le i \le n_d \\ 0 & \text{for} \quad n_{d+1} \le i \le n. \end{cases}$$

To solve the BDF-discretized NDAE system under Newton-Raphson we now implicitly represent the discrete-time model under NR iteration index(i). First, the implicit representation for the differential equation can we written as (41)

$$\boldsymbol{\phi}(\boldsymbol{z}_k^{(i)}) = \boldsymbol{x}_{d,k}^{(i)} - \sum_{s=1}^{k_g} \alpha_s \boldsymbol{x}_{d,k-s} - \tilde{h}\boldsymbol{f}(\boldsymbol{z}_k^{(i)}), \tag{41}$$

thus the BDF-discretized NDAE system (40) can be implicitly and succinctly written as (42)

$$\boldsymbol{0} = \boldsymbol{\phi}(\boldsymbol{z}_k^{(i)}) \tag{42a}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{x}_k^{(i)}). \tag{42b}$$

Following same methodology for simulating the system as with the μ-NDAE, the Jacobian $A_g(\cdot)$ of the implicit NDAE system (42) can be written as (43)

$$\begin{aligned} \boldsymbol{A}_g(\boldsymbol{z}_k^{(i)}) &= \begin{bmatrix} \frac{\partial \boldsymbol{\phi}(\boldsymbol{z}_k^{(i)})}{\partial \boldsymbol{x}_d} & \frac{\partial \boldsymbol{\phi}(\boldsymbol{z}_k^{(i)})}{\partial \boldsymbol{x}_a} \\ \\ \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k^{(i)})}{\partial \boldsymbol{x}_d} & \frac{\partial \boldsymbol{g}(\boldsymbol{x}_k^{(i)})}{\partial \boldsymbol{x}_a} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{I}_{n_d} - \tilde{h}\boldsymbol{F}_{x_d}(\boldsymbol{z}_k^{(i)}) & -\tilde{h}\boldsymbol{F}_{x_a}(\boldsymbol{z}_k^{(i)}) \\ \boldsymbol{G}_{x_d}(\boldsymbol{x}_k^{(i)}) & \boldsymbol{G}_{x_a}(\boldsymbol{x}_k^{(i)}) \end{bmatrix}, \end{aligned} \tag{43}$$

then for each iteration (i) the NR method increment is computed using equation (44), which is then used to update $\boldsymbol{x}^{(i+1)}$ in equation (45) until convergence is satisfied. Convergence is calculated using the $\mathcal{L}$-2–norm on the increment.

$$\Delta\boldsymbol{x}_k^{(i)} = [\boldsymbol{A}_g(\boldsymbol{z}_k^{(i)})]^{-1} \begin{bmatrix} \boldsymbol{\phi}(\boldsymbol{z}_k^{(i)}) \\ \boldsymbol{g}(\boldsymbol{x}_k^{(i)}) \end{bmatrix}, \tag{44a}$$

$$\boldsymbol{x}_{d,k}^{(i+1)} = \boldsymbol{x}_{d,k}^{(i)} + \Delta\boldsymbol{x}_{d,k}^{(i)}, \tag{45a}$$
$$\boldsymbol{x}_{a,k}^{(i+1)} = \boldsymbol{x}_{a,k}^{(i)} + \Delta\boldsymbol{x}_{a,k}^{(i)}. \tag{45b}$$

For brevity we only present the formulation under BDF discretization, we leave it for interested readers to infer the formulations for BE and TI methods from the presented work.

APPENDIX C
JACOBIAN MATRIX UNDER EXPLICIT REPRESENTATION

To calculate the Jacobian matrix for the nonlinear DAE vector valued function that represents the dynamics of the power system, we adopt a numerical layout notation. Then using the chain rule we express the Jacobian matrix $\boldsymbol{J}(\boldsymbol{x}_0)$ as shown in equation (23). Given the implicit discretization model used to simulate the system, we run into a challenge of explicitly representing $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$. This is due to the fact that $\boldsymbol{x}_k$ appears on both sides of the discrete system dynamics equations. We herein show the derivations and calculations required to obtain an explicit representation of $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$ for computing the Jacobian of the observability matrix $\boldsymbol{W}_o(\boldsymbol{\Gamma}, \boldsymbol{x}_0) \equiv \boldsymbol{W}_o^1(\mathcal{Z}, \boldsymbol{x}_0)$.

For Gear's discretization method, the partial derivative can be obtained by taking the partial derivative of the discretized system (40) as follows

$$\boldsymbol{E}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} = \boldsymbol{E}\sum_{s=1}^{k_g}\alpha_s\frac{\partial \boldsymbol{x}_{j-s}}{\partial \boldsymbol{x}_{j-1}} + \begin{bmatrix}\tilde{h} & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}\\ \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}\end{bmatrix}, \tag{46}$$

which can equivalently be written as

$$\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} = \alpha_{s(1)}\frac{\partial \boldsymbol{x}_{j-1}}{\partial \boldsymbol{x}_{j-1}} + \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} \tag{47a}$$
$$\boldsymbol{0} = \boldsymbol{0} + \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}. \tag{47b}$$

It is evident from (47) that $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$ for the case of differential variables in equation (47a) can be explicitly formulated assuming that $(\boldsymbol{I} + \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}})$ is invertable. However, for the algebraic constraint variables in equation (47b), $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$ is equal to zero and cannot be explicitly represented. This is where the μ-NDAE formulation comes into play to allow for a plausible solution towards developing an observability based sensor placement for descriptor systems of index-1.

To construct the Jacobian in (23) we represent the partial derivative $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$ explicitly for each of the discretization methods presented in this work. The partial derivative $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$ for each of the discretization method can be written as (48)

$$\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} := \begin{bmatrix}\frac{\partial \boldsymbol{x}_{d,j}}{\partial \boldsymbol{x}_{d,j-1}} & \frac{\partial \boldsymbol{x}_{d,j}}{\partial \boldsymbol{x}_{a,j-1}}\\ \frac{\partial \boldsymbol{x}_{a,j}}{\partial \boldsymbol{x}_{d,j-1}} & \frac{\partial \boldsymbol{x}_{a,j}}{\partial \boldsymbol{x}_{a,j-1}}\end{bmatrix}, \tag{48}$$

the partial derivative $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$ can be explicitly written for BDF method as per the following

$$\boldsymbol{E}_\mu\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} = \sum_{s=1}^{k_g}\alpha_s\boldsymbol{E}_\mu\frac{\partial \boldsymbol{x}_{j-s}}{\partial \boldsymbol{x}_{j-1}} + \tilde{h}\boldsymbol{I}_n\begin{bmatrix}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}\\ \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}\end{bmatrix}, \tag{49}$$

whereby differentiating with respect to the differential and algebraic state variables separately we obtain (50)

$$\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\alpha_{s(1)} + \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}} & \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}}\end{bmatrix} \tag{50a}$$

$$\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\tilde{h}\frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}} & \mu\alpha_{s(1)} + \tilde{h}\frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}}\end{bmatrix}, \tag{50b}$$

equation (50) is implicitly stated and here we can explicitly represent $\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}}$ and $\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}}$ as (51)

$$\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\left[\boldsymbol{I}_{n_d} - \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j}\right]^{-1}\alpha_{s(1)}\\ \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}}\end{bmatrix} \tag{51a}$$

$$\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\tilde{h}\frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j}\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}}\mu^{-1}\\ \left[\mu\boldsymbol{I}_{n_a} - \tilde{h}\frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j}\right]^{-1}\mu\alpha_{s(1)}\end{bmatrix}, \tag{51b}$$

let $\boldsymbol{A}_{n_d} = (\boldsymbol{I}_{n_d} - \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j})$ and $\boldsymbol{A}_{n_a} = (\mu\boldsymbol{I}_{n_a} - \tilde{h}\frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j})$, such that $\boldsymbol{A}_{n_d}$ and $\boldsymbol{A}_{n_a}$ are invertable. To explicitly represent $\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{a_{j-1}}}$ and $\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{d_{j-1}}}$, we replace $\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}}$ and $\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}}$ by their explicit representation formulated in (51)

$$\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\left[\boldsymbol{A}_{n_d}\right]^{-1}\alpha_{s(1)}\boldsymbol{I}_{n_d}\\ \tilde{h}\frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j}\left[\boldsymbol{A}_{n_a}\right]^{-1}\mu\alpha_{s(1)}\boldsymbol{I}_{n_a}\end{bmatrix} \tag{52a}$$

$$\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\tilde{h}\frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j}\left[\boldsymbol{A}_{n_d}\right]^{-1}\alpha_{s(1)}\boldsymbol{I}_{n_d}\mu^{-1}\\ \left[\boldsymbol{A}_{n_a}\right]^{-1}\mu\alpha_{s(1)}\boldsymbol{I}_{n_a}\end{bmatrix}, \tag{52b}$$

Now for the case where $k_g = 1$, that is the case of Backward Euler, (52) can be written as

$$\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\left[\boldsymbol{A}_{n_d}\right]^{-1}\boldsymbol{I}_{n_d} & \tilde{h}\boldsymbol{F}_{\boldsymbol{x}_{a,j}}\left[\boldsymbol{A}_{n_a}\right]^{-1}\mu\boldsymbol{I}_{n_a}\\ \tilde{h}\boldsymbol{G}_{\boldsymbol{x}_{d,j}}\left[\boldsymbol{A}_{n_d}\right]^{-1}\boldsymbol{I}_{n_d}\mu^{-1} & \left[\boldsymbol{A}_{n_a}\right]^{-1}\mu\boldsymbol{I}_{n_a}\end{bmatrix}. \tag{53}$$

where $\boldsymbol{F}_{\boldsymbol{x}_{d,j}} = \frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j}$, $\boldsymbol{F}_{\boldsymbol{x}_{a,j}} = \frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j}$, $\boldsymbol{G}_{\boldsymbol{x}_{d,j}} = \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{d_j}}|_{\boldsymbol{x}_j}$, and $\boldsymbol{G}_{\boldsymbol{x}_{a,j}} = \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_{a_j}}|_{\boldsymbol{x}_j^{(i)}}$.

To represent the partial derivative $\frac{\partial \boldsymbol{x}_{j+1}}{\partial \boldsymbol{x}_j}$ explicitly for the TI discrete time model of the system, we start by differentiating (12a) with respect to to both the differential and algebraic variables, and by denoting $\boldsymbol{F}_{\boldsymbol{x}_j} = \frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j}$ and $\boldsymbol{G}_{\boldsymbol{x}_j} = \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j^{(i)}}$. Differentiating with respect to the differential and algebraic state variables separately (12a) can be written as (54)

$$\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\boldsymbol{I}_{n_d} + \tilde{h}(\boldsymbol{F}_{\boldsymbol{x}_{d,j}}\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}} + \boldsymbol{F}_{\boldsymbol{x}_{d,j-1}})\\ \tilde{h}(\boldsymbol{F}_{\boldsymbol{x}_{a,j}}\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}} + \boldsymbol{F}_{\boldsymbol{x}_{a,j-1}})\end{bmatrix} \tag{54a}$$

$$\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{j-1}} = \begin{bmatrix}\tilde{h}(\boldsymbol{G}_{\boldsymbol{x}_{d,j}}\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}} + \boldsymbol{G}_{\boldsymbol{x}_{d,j-1}})\\ \mu\boldsymbol{I}_{n_a} + \tilde{h}(\boldsymbol{G}_{\boldsymbol{x}_{a,j}}\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}} + \boldsymbol{G}_{\boldsymbol{x}_{a,j-1}})\end{bmatrix}, \tag{54b}$$

then, the explicit representation of $\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}}$ and $\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{aj-1}}$ can be written as

$$\begin{bmatrix} \frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}} \\ \\ \frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{a_{j-1}}} \end{bmatrix} = \begin{bmatrix} \left[\boldsymbol{I}_{n_d} - \tilde{h}\boldsymbol{F}_{\boldsymbol{x}_{d,j}}\right]^{-1}\left[\boldsymbol{I}_{n_d} + \tilde{h}\boldsymbol{F}_{\boldsymbol{x}_{d,j-1}}\right] \\ \\ \tilde{h}(\boldsymbol{F}_{\boldsymbol{x}_{a,j}}\frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{a_{j-1}}} + \boldsymbol{F}_{\boldsymbol{x}_{a,j-1}}) \end{bmatrix} \tag{55a}$$

$$\begin{bmatrix} \frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{d_{j-1}}} \\ \\ \frac{\partial \boldsymbol{x}_{a_j}}{\partial \boldsymbol{x}_{aj-1}} \end{bmatrix} = \begin{bmatrix} \tilde{h}(\boldsymbol{G}_{\boldsymbol{x}_{d,j}}\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}} + \boldsymbol{G}_{\boldsymbol{x}_{d,j-1}})\mu^{-1} \\ \\ \left[\boldsymbol{\mu}_{n_a} - \tilde{h}\boldsymbol{G}_{\boldsymbol{x}_{a,j}}\right]^{-1}\left[\boldsymbol{\mu}_{n_a} + \tilde{h}\boldsymbol{G}_{\boldsymbol{x}_{a,j-1}}\right]\boldsymbol{\mu}_{n_a} \end{bmatrix}, \tag{55b}$$

where $\boldsymbol{\mu}_{n_a} := \mu \boldsymbol{I}_{n_a}$. Now, let for TI method, $\boldsymbol{A}^{-}_{n_{d,j}} = (\boldsymbol{I}_{n_d} - \tilde{h}\boldsymbol{F}_{\boldsymbol{x}_{d,j}})$, $\boldsymbol{A}^{+}_{n_{d,j}} = (\boldsymbol{I}_{n_d} + \tilde{h}\boldsymbol{F}_{\boldsymbol{x}_{d,j}})$, $\boldsymbol{A}^{-}_{n_{a,j}} = (\boldsymbol{\mu}_{n_a} - \tilde{h}\boldsymbol{G}_{\boldsymbol{x}_{a,j}})$ and $\boldsymbol{A}^{+}_{n_{a,j}} = (\boldsymbol{\mu}_{n_a} + \tilde{h}\boldsymbol{G}_{\boldsymbol{x}_{a,j}})$, such that $\boldsymbol{A}_{n_{a,j}}$ and $\boldsymbol{A}_{n_{a,j}}$ are invertable. Then, $\frac{\partial \boldsymbol{x}_{j+1}}{\partial \boldsymbol{x}_j}$ can be explicitly formulated and written for TI discretization as (56)

$$\left[\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}\right] = \begin{bmatrix} \left[\boldsymbol{A}^{-}_{n_{d,j}}\right]^{-1}\left[\boldsymbol{A}^{+}_{n_{d,j-1}}\right] \\ \\ \tilde{h}(\boldsymbol{F}_{\boldsymbol{x}_{a,j}}\left[\boldsymbol{A}^{-}_{n_{a,j}}\right]^{-1}\left[\boldsymbol{A}^{+}_{n_{a,j-1}}\right] + \boldsymbol{F}_{\boldsymbol{x}_{a,j-1}}) \\ \\ \tilde{h}(\boldsymbol{G}_{\boldsymbol{x}_{d,j}}\left[\boldsymbol{A}^{-}_{n_{d,j}}\right]^{-1}\left[\boldsymbol{A}^{+}_{n_{d,j-1}}\right] + \boldsymbol{G}_{\boldsymbol{x}_{d,j-1}})\mu^{-1} \\ \\ \left[\boldsymbol{A}^{-}_{n_{a,j}}\right]^{-1}\left[\boldsymbol{A}^{+}_{n_{a,j-1}}\right] \end{bmatrix}, \tag{56}$$

the partial derivative $\frac{\partial \boldsymbol{x}_{d_j}}{\partial \boldsymbol{x}_{d_{j-1}}}$ is now explicitly defined, as such we can now concatenate the Jacobian $\boldsymbol{J}(\cdot)$ for observation horizon N_o.

Appendix D
Partial Derivatives for Gear's discretization

To capture the system dynamics for Gear's discretization of order k_g, we realize that the formulation for the partial differential $\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}}$ is not fully representative for BDF of order k_g except when $k_g = 1$ for reasons that will be obvious shortly. On the other hand, building the Jacobian by differentiating the states with respect to k_g will allow to fully capture the system dynamics. Herein we show how this formulation captures the full system dynamics by taking Gear's discretization order $k_g = 3$ as an example. Differentiating the system dynamics with with respect to $\boldsymbol{x}_{j-1}$ for $k_g = 3$ yields the following

$$\begin{aligned} \boldsymbol{E}_\mu \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} =& \boldsymbol{E}_\mu(\alpha_{(1)} \underbrace{\frac{\partial \boldsymbol{x}_{j-1}}{\partial \boldsymbol{x}_{j-1}}}_{\boldsymbol{I}_n} + \alpha_{(2)} \underbrace{\frac{\partial \boldsymbol{x}_{j-2}}{\partial \boldsymbol{x}_{j-1}}}_{0} + \alpha_{(3)} \underbrace{\frac{\partial \boldsymbol{x}_{j-3}}{\partial \boldsymbol{x}_{j-1}}}_{0}) \\ &+ \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j} \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} \\ \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j} \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-1}} \end{bmatrix}. \end{aligned} \tag{57}$$

From (57) we notice that if we differentiate the system dynamics with respect to one prior time step $(j-1)$ and when the order of k_g is greater than 1, the system dynamics depicted in the $\sum_{s=2}^{k_g} \alpha_s \frac{\partial \boldsymbol{x}_{j-s}}{\partial \boldsymbol{x}_{j-1}}$ suffice equal to zero. Utilizing such approach, the BDF method for any order between $1 \leq k_g \leq 5$ will represent only that of order 1.

Realizing such shortcoming, we refer to considering Gear's discretization order for developing the partial derivatives. As such, we differentiate the discretized system (40) with respect to k_g in order to capture the full system dynamics. For $k_g = 3$ the partial derivative can be written as (58)

$$\begin{aligned} \boldsymbol{E}_\mu \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-3}} =& \boldsymbol{E}_\mu(\alpha_{(1)} \frac{\partial \boldsymbol{x}_{j-1}}{\partial \boldsymbol{x}_{j-3}} + \alpha_{(2)} \frac{\partial \boldsymbol{x}_{j-2}}{\partial \boldsymbol{x}_{j-3}} + \alpha_{(3)} \underbrace{\frac{\partial \boldsymbol{x}_{j-3}}{\partial \boldsymbol{x}_{j-3}}}_{\boldsymbol{I}_n}) \\ &+ \tilde{h}\boldsymbol{I}_n \begin{bmatrix} \frac{\partial \boldsymbol{f}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j} \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-3}} \\ \frac{\partial \boldsymbol{g}(\boldsymbol{x}_j)}{\partial \boldsymbol{x}_j}|_{\boldsymbol{x}_j} \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-3}} \end{bmatrix}, \end{aligned} \tag{58}$$

where $\frac{\partial \boldsymbol{x}_{j-1}}{\partial \boldsymbol{x}_{j-3}} = \frac{\partial \boldsymbol{x}_{j-1}}{\partial \boldsymbol{x}_{j-2}} \frac{\partial \boldsymbol{x}_{j-2}}{\partial \boldsymbol{x}_{j-3}}$.

Taking $j = k_g = 3$, we now have defined $\frac{\partial \boldsymbol{x}_3}{\partial \boldsymbol{x}_0}$ as (58) and $\frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0}$. Such that $\frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1}$ and $\frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0}$ are calculated using the formulation presented in (52).

Now, for $j \geq 2k_g$ that is in this case $j \geq 6$, we use the chain rule to compute the j-th derivative with respect to $\boldsymbol{x}_0$ as follows

$$\frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}_j}{\partial \boldsymbol{x}_{j-k_g}} \cdots \frac{\partial \boldsymbol{x}_{k_g}}{\partial \boldsymbol{x}_0}, \tag{59}$$

and for $k_g < j < 2k_g$ that is in this case $3 < j < 6$, we represent the j-th derivative for $j = 4$ and $j = 5$ with respect to $\boldsymbol{x}_0$ respectively as follows

$$\frac{\partial \boldsymbol{x}_4}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}_4}{\partial \boldsymbol{x}_1} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0} \tag{60a}$$

$$\frac{\partial \boldsymbol{x}_5}{\partial \boldsymbol{x}_0} = \frac{\partial \boldsymbol{x}_5}{\partial \boldsymbol{x}_2} \frac{\partial \boldsymbol{x}_2}{\partial \boldsymbol{x}_1} \frac{\partial \boldsymbol{x}_1}{\partial \boldsymbol{x}_0}. \tag{60b}$$

Such that, $\frac{\partial \boldsymbol{x}_5}{\partial \boldsymbol{x}_2}$ and $\frac{\partial \boldsymbol{x}_4}{\partial \boldsymbol{x}_1}$ are calculated from (58). Given such formulation, the Jacobian for Gear's discretization method is able to depict system dynamics for $k_g > 1$.

References

[1] C. Zhao, U. Topcu, and S. H. Low, "Frequency-based load control in power systems," *Proceedings of the American Control Conference*, pp. 4423–4430, 2012.

[2] T. Chen, H. Ren, E. Y. S. Foo, L. Sun, and G. A. J. Amaratunga, "A Fast and Robust State Estimator Based on Exponential Function for Power Systems," *IEEE Sensors Journal*, pp. 1–1, jan 2022.

[3] A. Ahmadi, Y. Alinejad-Beromi, and M. Moradi, "Optimal PMU placement for power system observability using binary particle swarm optimization and considering measurement redundancy," *Expert Systems with Applications*, vol. 38, no. 6, pp. 7263–7269, jun 2011.

[4] W. Yuill, A. Edwards, S. Chowdhury, and S. P. Chowdhury, "Optimal PMU placement: A comprehensive literature review," in *IEEE Power and Energy Society General Meeting*, 2011.

[5] J. Qi, K. Sun, and W. Kang, "Optimal PMU Placement for Power System Dynamic State Estimation by Using Empirical Observability Gramian," *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 2041–2054, jul 2015.

[6] K. Sun, J. Qi, and W. Kang, "Power system observability and dynamic state estimation for stability monitoring using synchrophasor measurements," *Control Engineering Practice*, vol. 53, pp. 160–172, 2016.

[7] S. A. Nugroho and A. F. Taha, "Towards understanding sensor and control nodes selection in nonlinear dynamic systems: Lyapunov theory meets branch-and-bound," *Automatica*, vol. 134, p. 109904, 2021.

[8] M. Ali Abooshahab, M. Hovd, and G. Valmorbida, "Optimal Sensor Placement for Partially Known Power System Dynamic Estimation," *Proceedings of 2021 IEEE PES Innovative Smart Grid Technologies Europe: Smart Grids: Toward a Carbon-Free Future, ISGT Europe 2021*, 2021.

[9] N. M. Manousakis, G. N. Korres, and P. S. Georgilakis, "Taxonomy of PMU placement methodologies," *IEEE Transactions on Power Systems*, vol. 27, no. 2, pp. 1070–1077, 2012.

[10] M. Nazari-Heris and B. Mohammadi-Ivatloo, "Application of heuristic algorithms to optimal PMU placement in electric power systems: An updated review," *Renewable and Sustainable Energy Reviews*, vol. 50, pp. 214–228, 2015.

[11] G. J. Balas and P. M. Young, "Sensor selection via closed-loop control objectives," *IEEE Transactions on Control Systems Technology*, vol. 7, no. 6, pp. 692–705, 1999.

[12] T. H. Summers, F. L. Cortesi, and J. Lygeros, "On Submodularity and Controllability in Complex Dynamical Networks," *IEEE Transactions on Control of Network Systems*, vol. 3, no. 1, pp. 91–101, mar 2016.

[13] A. Haber, F. Molnar, and A. E. Motter, "State Observation and Sensor Selection for Nonlinear Networks," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 2, pp. 694–708, 2018.

[14] A. Rouhani and A. Abur, "Observability Analysis for Dynamic State Estimation of Synchronous Machines," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3168–3175, 2017.

[15] A. P. Vinod, A. J. Thorpe, P. A. Olaniyi, T. H. Summers, and M. M. Oishi, "Sensor Selection for Dynamics-Driven User-Interface Design," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 1, pp. 71–84, 2022.

[16] C. Letellier, L. A. Aguirre, and J. Maquet, "Relation between observability and differential embeddings for nonlinear dynamics," *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, vol. 71, no. 6, pp. 1–8, 2005.

[17] A. J. Whalen, S. N. Brennan, T. D. Sauer, and S. J. Schiff, "Observability and controllability of nonlinear networks: The role of symmetry," *Physical Review X*, vol. 5, no. 1, pp. 1–18, 2015.

[18] S. Lall, J. E. Marsden, and S. Glavaski, "Empirical model reduction of controlled nonlinear systems," *International Federation of Automatic Control*, pp. 2598–2603, 1999.

[19] J. Hahn and T. F. Edgar, "Balancing approach to minimal realization and model reduction of stable nonlinear systems," *Industrial and Engineering Chemistry Research*, vol. 41, no. 9, pp. 2204–2212, 2002.

[20] A. J. Krener and K. Ide, "Measures of unobservability," *Proceedings of the IEEE Conference on Decision and Control*, pp. 6401–6406, 2009.

[21] M. Serpas, G. Hackebeil, C. Laird, and J. Hahn, "Sensor location for nonlinear dynamic systems via observability analysis and MAX-DET optimization," *Computers and Chemical Engineering*, vol. 48, pp. 105–112, 2013.

[22] J. Qi, K. Sun, and W. Kang, "Adaptive Optimal PMU Placement Based on Empirical Observability Gramian," *IFAC-PapersOnLine*, vol. 49, no. 18, pp. 482–487, 2016.

[23] I. Yang, S. A. Burden, R. Rajagopal, S. S. Sastry, and C. J. Tomlin, "Approximation Algorithms for Optimization of Combinatorial Dynamical Systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 9, pp. 2644–2649, 2016.

[24] A. A. Saleh, A. S. Adail, and A. A. Wadoud, "Optimal phasor measurement units placement for full observability of power system using improved particle swarm optimisation," *IET Generation, Transmission and Distribution*, vol. 11, no. 7, pp. 1794–1800, 2017.

[25] L. Romao, K. Margellos, and A. Papachristodoulou, "Distributed Actuator Selection: Achieving Optimality via a Primal-Dual Algorithm," *IEEE Control Systems Letters*, vol. 2, no. 4, pp. 779–784, 2018.

[26] Z. Zheng, Y. Xu, L. Mili, Z. Liu, M. Korkali, and Y. Wang, "Observability Analysis of a Power System Stochastic Dynamic Model Using a Derivative-Free Approach," *IEEE Transactions on Power Systems*, vol. 36, no. 6, pp. 5834–5845, 2021.

[27] L. Dang, W. Wang, and B. Chen, "Square Root Unscented Kalman Filter With Modified Measurement for Dynamic State Estimation of Power Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, 2022.

[28] K. Manohar, J. N. Kutz, and S. L. Brunton, "Optimal Sensor and Actuator Selection Using Balanced Model Reduction," *IEEE Transactions on Automatic Control*, vol. 67, no. 4, pp. 2108–2115, 2022.

[29] Y. Liu and K. Sun, "Solving Power System Differential Algebraic Equations Using Differential Transformation," *IEEE Transactions on Power Systems*, vol. 35, no. 3, pp. 2289–2299, may 2020.

[30] S. A. Nugroho, A. Taha, N. Gatsis, and J. Zhao, "Observers for Differential Algebraic Equation Models of Power Networks: Jointly Estimating Dynamic and Algebraic States," *IEEE Transactions on Control of Network Systems*, vol. 5870, no. c, 2022.

[31] T. Groß, S. Trenn, and A. Wirsen, "Solvability and stability of a power system DAE model," *Systems and Control Letters*, vol. 97, pp. 12–17, 2016.

[32] M. H. Kazma and A. F. Taha, "Optimal Placement of PMUs in Power Networks: Modularity Meets A Priori Optimization," *Proceedings of the American Control Conference*, vol. 2023-June, pp. 4489–4494, 2023.

[33] P. W. Sauer, M. A. Pai, and J. H. Chow, *Power System Dynamics and Stability: With Synchrophasor Measurement and Power System Toolbox 2e*, 1st ed. Wiley and Sons Ltd, 2017.

[34] L. F. Shampine, M. W. Reichelt, and J. A. Kierzenka, "Solving index-I DAEs in MATLAB and Simulink," *SIAM Review*, vol. 41, no. 3, pp. 538–552, 1999.

[35] J. Y. Astic, M. Jerosolimski, and A. Bihain, "The mixed adams - BDF variable step size algorithm to simulate transient and long term phenomena in power systems," *IEEE Transactions on Power Systems*, vol. 9, no. 2, pp. 929–935, 1994.

[36] C. W. Gear, "Simultaneous Numerical Solution of Differential-Algebraic Equations," *IEEE Transactions on Circuit Theory*, vol. 18, no. 1, pp. 89–95, 1971.

[37] U. M. Ascher and L. R. Petzold, "Projected Implicit Runge-Kutta methods for differential-algebraic equations," *SIAM Journal on Numerical Analysis*, vol. 28, no. 4, pp. 1097–1120, 1991.

[38] E. E. Hairer, *Solving ordinary differential equations*, 2nd ed., ser. Springer series in computational mathematics ; 8,14. Berlin ;: Springer-Verlag, 1993.

[39] F. A. Potra, M. Anitescu, B. Gavrea, and J. Trinkle, "A linearly implicit trapezoidal method for integrating stiff multibody dynamics with contact, joints, and friction," *International Journal for Numerical Methods in Engineering*, vol. 66, no. 7, pp. 1079–1124, 2006.

[40] F. Milano, "Semi-Implicit Formulation of Differential-Algebraic Equations for Transient Stability Analysis," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 4534–4543, 2016.

[41] T. B. Gross, S. Trenn, and A. Wirsen, "Topological solvability and index characterizations for a common DAE power system model," *2014 IEEE Conference on Control Applications, CCA. Part of 2014 IEEE Multi-conference on Systems and Control, MSC 2014*, pp. 9–14, 2014.

[42] S. G. Krantz and H. R. Parks, *The Implicit Function Theorem*. Springer New York, 2013.

[43] M. Abdelmoula, S. Moughamir, and B. Robert, "Analysis and implementation of nonlinear implicit differential-algebraic equations solver: Application to a photovoltaic power system," *13th International Multi-Conference on Systems, Signals and Devices, SSD 2016*, no. 1, pp. 735–741, 2016.

[44] S. Kazuhiro, "With Geometric Index One," in *52nd IEEE Conference on Decision and Control*. Firenze, Italy: IEEE, 2013, pp. 2582–2587.

[45] M. Hou and P. C. Müller, "Observer design for a class of descriptor systems," *European Control Conference, ECC 2003*, no. September, pp. 3005–3010, 2003.

[46] F. Milano, Ed., *Advances in Power System Modelling, Control and Stability Analysis*, 2nd ed. London: IET, 2022.

[47] A. Alessandri, M. Baglietto, and G. Battistelli, "Moving-horizon state estimation for nonlinear discrete-time systems: New stability results and approximation schemes," *Automatica*, vol. 44, no. 7, pp. 1753–1765, 2008.

[48] S. A. Nugroho, S. C. Vishnoi, A. F. Taha, and C. G. Claudel, "Where Should Traffic Sensors Be Placed on Highways ?" *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 1–14, 2021.

[49] R. D. Zimmerman, C. E. Murillo-sánchez, and R. J. Thomas, "MATPOWER : Steady-State Operations , Systems Research and Education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[50] S. Hanba, "On the "Uniform" Observability of Discrete-Time Nonlinear Systems," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1925–1928, 2009.

[51] A. Minot, Y. M. Lu, and N. Li, "A Distributed gauss-Newton method for power system state estimation," *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3804–3815, 2016.

[52] J. Löfberg, "YALMIP : A Toolbox for Modeling and Optimization in MATLAB," in *In Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

[53] L. Gurobi Optimization, "Gurobi Optimizer Reference Manual," 2022.

[54] L. Lovász, *Submodular functions and convexity*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 235–257.

[55] G. Calinescu, C. Chekuri, and J. Pàl, Martin; Vondrák, "Maximizing a Monotone Submodular Function Subject to a Matroid Constraint," *SIAM Journal of Computing*, vol. 40, no. 6, pp. 1740–1766, 2011.

[56] H. Maros and S. Juniar, "A Priori Optimization," *Operations Research*, vol. 38, no. 6, pp. 1–23, 1990.

[57] A. B. Birchfield, K. M. Gegner, T. Xu, K. S. Shetye, and T. J. Overbye, "Statistical Considerations in the Creation of Realistic Synthetic Power Grids for Geomagnetic Disturbance Studies," *IEEE Transactions on Power Systems*, vol. 32, no. 2, pp. 1502–1510, 2017.